EVALUATION OF CONCEPTUAL SKETCHES ON STYLUS-BASED DEVICES

A Thesis

by

SHALINI PRIYA ASHOK KUMAR

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Tracy Hammond
Committee Members,	Frank Shipman
	Richard R. Davison
Head of Department,	Dilma Di Silva

May 2016

Major Subject: Computer Science and Engineering

ABSTRACT

Design Sketching is an important tool for designers and creative professionals to express their ideas and thoughts onto visual medium. Being a very critical and versatile skill for engineering students, this course is often taught in universities on pen and paper. However, this traditional pedagogy is limited by the availability of human instructors for their feedback. Also, students having low self-efficacy do not learn efficiently in traditional learning environment.

Using intelligent interfaces this problem can be solved where we try to mimic the feedback given by an instructor and assess the student drawn sketches to give them insight of the areas they need to improve on. PerSketchTivity is an intelligent tutoring system which allows students to practice their drawing fundamentals and gives them real-time assessment and feedback. This research deals with finding the evaluation metrics that will enable us to grade students from their sketch data. There are seven metrics that we will work with to analyse how each of them contribute in deciding the quality of the sketches. The main contribution of this research is to identify the features of the sketch that can distinguish a good quality sketch from a poor one and design a grading metric for the sketches that can give a final score between 0 and 1 to the user sketches. Using these obtained features and our grading metric method, we grade all the sketches of students and experts.

Dedicated to my parents, brother and Gautham

ACKNOWLEDGEMENTS

I would like to express my sincere thanks to everyone who helped in the completion of this thesis. My gratitude to my advisor Dr. Tracy Hammond for constant support throughout the Masters and inspiring and motivating me. Her continuous guidance, mentoring and encouragement helped me learn and progress in the research.

I also would like to thank other committee members Dr. Frank Shipman and Prof. Richard R. Davison for insightful suggestions, sharing their knowledge and helping out in finding the experts in the field.

I am grateful to the PerSketchTivity team at Georgia Institute of Technology for their insights during the research and helping in collecting data. My special thanks to Dr. Julie Linsey, Dr. Wayne Li, Blake Willford and Ethan Hilton without who implemented the software in their classroom to get student data.

Thanks to all the friends in the Sketch Recognition Lab for their valuable feedback throughout my Masters. Particularly I would like to thank Paul Taele and Swarna Keshavabotla who were part of the project and have given their comments during the course of research. My thanks to Stephanie Valentine, Seth Polsley, Jung-In Koh, Trevor Nelligan and Vijay Rajanna.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

Drawing is one of the earliest forms of human expression, predating written communication [91]. Throughout history it has been used in a wide range of fields including art, design, engineering, science and education. Learning to draw is considered vital in learning to produce other forms of visual art. Not only the skills acquired through drawing is helpful, but is often the first step in producing new artwork. It is fast, direct and can be done easily anywhere. The necessary materials are also cheap and extensively available. Drawing is very popular and used in many disciplines because of these reasons. An artist's ability and artistic vision can be revealed through their drawing.

Sketch is a rapidly executed rough, freehand drawing that is usually intended to be a basis for the final work [19]. The sketches are produced as preliminary drawing before creating a more sophisticated art work. They are usually drawn quickly with minimum details lacking very minute and tiny details. They are used to express and document ideas. It is one of the very important and commonly used forms of visual expression in design process for engineers. Designers and other creative professionals use design sketching to quickly generate and explore ideas and also communicate them to orders [47]. Even with growing popularity of modern computing devices and software, design sketching is used in various stages of design process by experts in the field of art, engineering and design [92] because of its benefits: convey and discuss preliminary design ideas [76], the hidden challenges and get solutions to the problems using visual sketches [76], assisting in ideation process during exploration of ideas with loose sketches [9], and engage the audience and attract them with designs [82]. They not only allow the designers to communicate their ideas, but let the learners improve their general academic performance and problem-solving capability [76], honing analytical skills [23], stimulating both halves of the brain [78], improving skills in writing and critical thinking when integrating sketching into their thought process [18]. Design sketching also assists in other academic areas by boosting self-confidence from successful artistic pursuits [53] and improves three-dimensional spatial recognition skills [83].

Due to the advantages of design sketching and its numerous applications in the field of engineering, exclusive design sketching courses have been devised and the students are encouraged to enroll for it. Traditionally, these courses are taught in classroom environments where students are taught by instructors and then students practice on their own which is later presented to their peers and instructor to get feedback. But it gets difficult for instructor to give valuable feedback as the class size increases. This work is also time consuming for humans.

To eliminate the problems that exist in the regular classroom setup, we have built an intelligent tutoring system, PerSketchTivity, which is an online stylus and touch capable interface where students can practice their design sketching fundamentals and get real-time feedback and assessment. PerSketchTivity has various lessons and challenges allowing students to practice basics of design sketching. One of the major challenges of such an application is to mimic the human instructors and to be able to give immediate and useful feedback about the sketches.

Feedback is necessary to assess the progress in the task of learning. Chen et al. [10] identifies immediate feedback as one of the important factors to have a good optimal experience for web activities. Students get motivated getting feedback immediately and learn the task faster and better attaining a mastery of the skill. It also keeps students engaged in the activity of learning. Stylus devices can be designed to give immediate feedback as if a human instructor was grading them by designing artificial intelligence algorithms.

Artificial intelligence not only can learn from given data but also has the capability of discovering new facts about the data [81]. In this thesis, we apply artificial intelligence on the sketch data to grant the application the power to grade the sketches automatically without human intervention. This will also allow us to discover the features that contribute to the good quality of sketches. Features will be extracted from the sketches to find out what fraction of each of these features decide the quality of sketches according to design sketching rubrics of experts.

The major contributions of this thesis are to identify if a computer can tell if a sketch is of good quality or not and to identify specific features of the sketch that make a sketch good. We have designed an evaluation metric based on the features of sketch to grade the sketches automatically. This helps in students not having to depend on anyone to get feedback about their sketches and learn the concepts of design sketching themselves and understand what it takes to draw a good design sketch.

In the remaining chapters, we discuss the motivation, approaches and contributions of this thesis more elaborately. The organization of the remaining chapters is as follows. Chapter two discusses the relevant work in the related fields. Since, PerSketchTivity is a web tool to teach students, we give a high level review of the works in the area of Intelligent tutoring system, mobile drawing applications and sketch recognition. In chapter three, we present details about concept or design sketches and traditional pedagogy of teaching these concepts in class. Chapter four gives a brief overview of PerSketchTivity, preprocessing of sketches and the recognition algorithms used. Chapter five focuses on development of the grading rubric and chapter six gives the details about evaluation of the system developed. Chapter seven discusses the evaluation system. We conclude the thesis with chapter eight and the last chapter provides the discussion of future work.

2. RELATED WORK

With the development and emergence of newer computing technologies, a lot of researchers and software developers have focused on using the technologies in fields related to sketching. This is true with advancements in hardware devices that can support digital sketching such as pen based devices, mobile tablet and touchscreen computers. Web-based educational applications and intelligent-tutoring system is another area that is gaining popularity in the domain of education and has proved to be an effective way of teaching and learning. Before looking into the details of the thesis, this section describes the related and relevant prior work to understand how PerSketchTivity and its feedback is different from existing systems.

2.1 Mobile Drawing Applications

Mobile based pen and stylus interface have developed in recent years and have been used widely in fields of drawing and sketching. There are numerous education drawing applications that exist today that teach design sketching. They can be categorised into two: Drawing instruction applications and Drawing gaming applications. While drawing instruction applications have adapted aspects of drawing instructions on mobile devices, drawing gaming applications have gamified the drawing instructions.

2.1.1 Drawing Instruction Applications

Learn to Draw Sketchbook by Walter Foster [63] is a drawing instruction application which focuses on fine art style of drawing and relies on step-by-step tracing approach to teach drawing. The step-by-step approach is effective, however it relies purely on tracing and does not teach to draw from imagination and perspective sketching. There is no way to gauge the performance of the users' sketch in the application. Figure 2.1 shows the screenshot of the application.

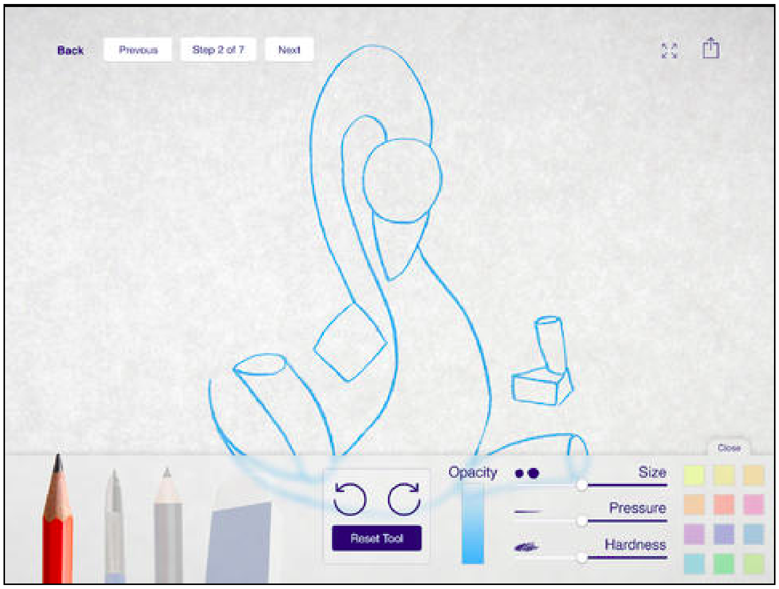

Figure 2.1: Learn to Draw Sketchbook application [63]

Draw This App from Peter Hamilton [49] is another application which uses step-by-step guidance approach. Another valuable feature that it incorporates is providing accuracy metrics on tracing shapes as shown in figure 2.2. The downside of this application is that it relies purely on tracing of shapes and forces users to draw shapes in a certain way that is not the proper technique used in design sketching. It does not help users understand 3-dimensionality and perspective. The assessment capability provided by this application is also very limited which does not help students understand the areas to focus on.

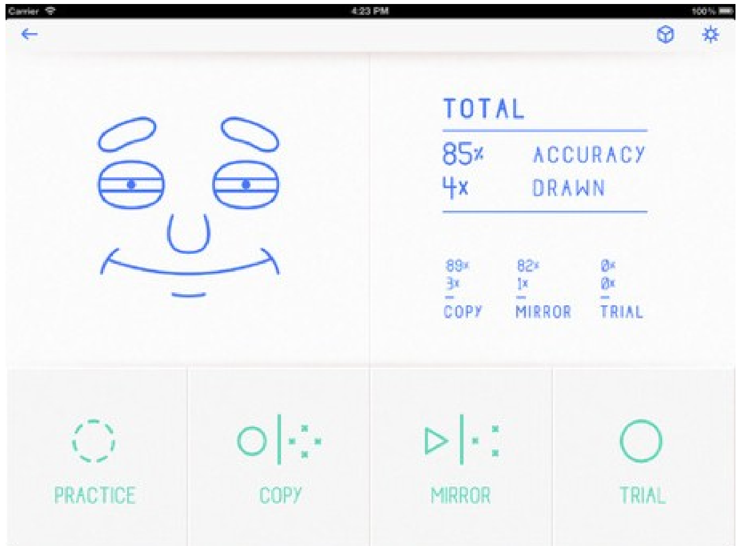

Figure 2.2: Draw This App [49]

2.1.2 Drawing Gaming Applications

Circled [85] developed by Underbeak [1] is an application in which the sole objective is to draw circles of various sizes accurately. It measures the accuracy of user's circles and depending on the performance, more levels and modes are unlocked. Figure 2.3 shows a screenshot of the application with the accuracy feedback. Though, this application provides feedback it is limited to one feature - accuracy of the circle. It does not consider the speed and the technique used while drawing circles which are important factors to label them as good sketches.

Another application called Draw Something [68] is a multiplayer game that allows people to sketch and have others guess what they sketched. This game can prove to be a very good tool to improve visual communication skill. That being said, it does not teach users any sketching technique and is not professionally oriented. This application also has no way to assess the sketches drawn.

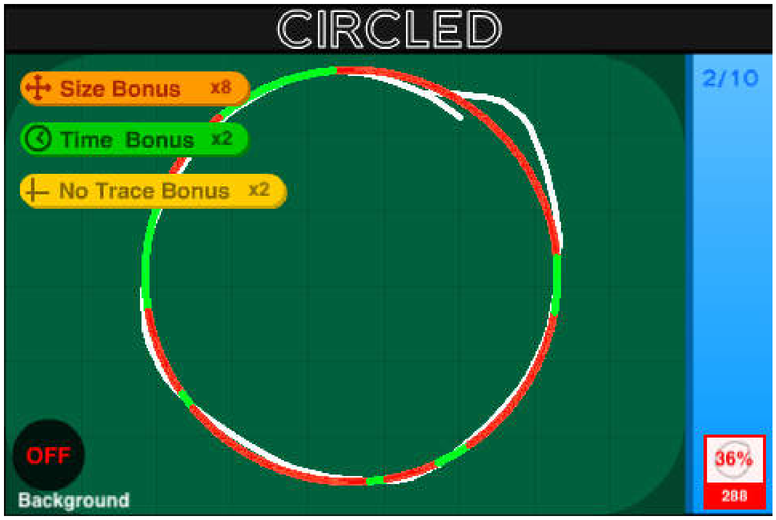

Figure 2.3: Circled application [85]

2.2 Intelligent Tutoring System

Another category of relevant research that relates to our study is Sketch-based Intelligent Tutoring system(ITS) in the field of sketching as well as other domains [73]. These applications, made for both children [97, 70] and adults, enable users to draw on sketch-based user interface and give them both feedback and assessment on their solutions. Mechanix [95, 72, 4, 3, 93, 96, 62, 25, 66, 94, 56] which is a sketch-based tutoring system for engineering students learning statics allows students to enter free body diagrams, specifically truss, into the system which are then checked against the solution entered by instructor to give detailed real-time feedback. Mechanix helps students reach the correct answer by giving beneficial feedback such as You are missing an input force at Node B and You have not drawn an axis. However, it does not grade the students' submissions and this still has to be done manually by the instructors.

Maestoso [87] is another tutoring system which uses sketch-based interface in the field of music for novice learners to learn music theory and enables students to progress through lessons covering fundamentals of music theory. This research tool also gives automated instructor-emulated feedback and review giving details about the corrections to their solution and also where students went wrong in their solution. Other representative disciplines where sketch-based tutoring systems has been incorporated include East Asian Languages [86, 88, 89, 90], Maths [54, 75], electrical engineering [20], coding [40, 42], and the military [46, 43, 14, 41, 16].

Tutoring system that are more closely related to drawing and sketching instruction have focused on figure art drawing. There are also many tutoring systems related to drawing and sketching instructions that focus on figure art drawing apart from mobile applications mentioned in previous section. Applications like iCanDraw [21, 45] and EyeSeeYou [15] provide direction and feedback to the users while drawing faces and eyes, respectively, from an image to learn drawing technique. Other art systems focus on instilling drawing confidence [61]. The Drawing Assistant [51], PortraitSketch [104] and Painting with Bob [6] are the extensions of iCanDraw and EyeSeeYou which implement a wider set of figures, detailed features or provide digital painting platform for novices. Though all of these systems focus on improving drawing style and skills of the novice users, there is no application that focuses on improving other elements needed in design sketching.

2.3 Sketch Recognition

Recognition of hand drawn sketches is called sketch recognition. The literature contains a great extent of research in the field of sketch recognition[33]. The sketch recognition algorithms can be classified primarily into three categories: Gesture-based recognition [77, 99, 59, 17, 11, 12], Geometric recognition [71, 30] and vision recognition [28] [65, 74].

Gesture based sketch recognition uses the inherent properties of the sketch to identify shapes. In these recognition algorithms either the system learns the user's style of drawing or the user has to learn in a style required by the system. Rubine used features like initial angle, sharpness, speed and total angle traversed to recognize shapes. These are some of the most popular features cited popularly in Sketch Recognition research and can even be used to predict the shape before it is completed [60]. Sezgin [80] and Staovich [84] took advantage of gesture based features like speed and curvature to distinguish different shapes.

Geometric recognizers usually do a bottom-up approach where the basic shapes like lines, arcs or circles are recognized first [44] and strokes are segmented into their components at corners [100, 103, 101, 102]. A higher level recognizer is built on top of this low-level recognizer which uses geometric constraints [52] to check if the primitive shapes when put together form a more complex shape [32, 38, 34, 39, 37, 31, 36, 35, 30]. They allow users to draw a shape in a natural way. PaleoSketch [71] is a very powerful low-level geometric recognizer which has capability of recognizing more than 10 basic shapes.

Vision based algorithms uses concepts from computer vision similar to those used on images after preprocessing of the sketches. The screen coordinates are used by Kara and Stahovich [55] to apply template matching algorithms used in their recognizer. Hausdorff, Modified Hausdorff, Tanimoto coefficient and Yule coefficient are used in the paper.

All the existing algorithms, or their combination [13] enable us to only identify different shapes whereas PerSketchTivity needs to assess the sketches as well. PerSketchTivity uses geometric recognition in its first phase to recognize shapes and gesture based features which help in distinguishing users for the evaluation or assessment phase. We take advantage of the fact that gesture based recognizers are user-dependent in our second phase of the system.

3. DESIGN SKETCHING

Before discussing the software and our methods of evaluation, this chapter gives an insight about design sketching, traditional pedagogy to teach design sketching and the actual process of sketching. This is helpful in understanding the motivation of PerSketchTivity and the evaluation system presented in this thesis.

3.1 Design Sketches

Sketching is an integral part of life of Industrial designers and engineers [92]. Design sketches or concept sketches are used in various stages of sketching. Sketches are not only used to document the ideas conceived in one's mind but also helps in acting as a stimuli for generating more ideas. It allows a designer to brainstorm more before getting to a final design. Sketches have become an essential process in the primary and initial stages of design and ideation for various disciplines such as architecture and engineering. Figure 3.1 is an example of design sketch of a camera.

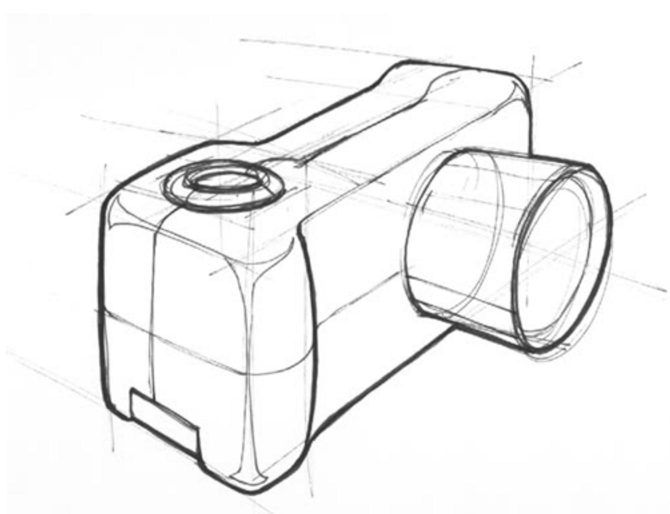

Figure 3.1: Design sketch of camera

3.2 Traditional Pedagogy

Students in many fields including industrial design, architecture, mechanical engineering and beyond academic programs are encouraged and required to take up the courses that teach fundamentals of design because of the benefits of design sketching and the related applications. These courses train students to learn sketching and draw quality sketches that reach the expertise level of the professionals in their respective disciplines by teaching them basics and different techniques involved in drawing good sketches.

Students have several resources accessible to them to develop skills in design sketching and master it. One of these resources is classroom setup which is often taught in studio environments. In this kind of set up students are taught by the instructors and they are expected to practice on their own. Later, these sketches are shared in front of their peers for their feedback. Classroom setting is the most prevalent approach for students in the universities where they receive direct feedback about the progress from their peers and experienced instructors. But as the classroom size increases it becomes very difficult for the instructors to give valuable feedback to all the students looking at each of their sketch. Also, it is not only a very time consuming task to give feedback individually to students but also the availability of instructors for feedback is limited to classroom and office hours [22].

One other problem associated with classroom environment where students need to share their work with their peers is that the students having low self-efficacy and level of self-confidence to complete a task have less to no motivation to participate and complete the tasks assigned to them in the classroom, thus hindering their learning. Since sketch is also affected by the self-efficacy, learning design sketching is no different. Students with low self-efficacy are less confident of improvement, not motivated to practice and are discouraged by more skilled peers with their drawings. It is a major challenge to raise the levels of self-efficacy to improve their levels of achievement and engagement [48].

Sketchbook is another popular and dominant supporting tool used by a lot of people to learn design sketching both outside of the classroom environment and in academic curriculum. Sketchbooks have proven to be an effective tool to learn sketching as they provide both portability and accessibility for students to practice their design sketching regularly and continuously [69]. Another advantage in using sketchbooks is that one can keep track of the history of the practice and progress which is important in learning.

3.3 What are Sketches?

In the previous sections, we understood where design sketches are used and how they are traditionally taught. This sections deals explains what sketches are and what are the qualities of a good sketch. Buxton provides in-depth knowledge of the characteristics of a design sketch [9]. He defines sketches as having the following qualities:

- Quick - A sketch is quick to make, or at least gives that impression.
- Timely - A sketch can be provided when needed.
- Inexpensive - A sketch is cheap. Cost must not inhibit the ability to explore a concept, especially early in the design process.
- Disposable - If you can't afford to throw it away when done, it is probably not a sketch.
- Plentiful - Sketches tend not to exist in isolation.

- Clear Vocabulary - The style in which a sketch is rendered follows certain conventions that distinguish it from other types of renderings.
- Distinct Gesture - There is a fluidity to sketches that give them a sense of openness and freedom.
- Minimal Detail - Include only what is required to render the intended purpose or concept.
- Appropriate Degree of Refinement - By its resolution or style, a sketch should not suggest a level of refinement beyond that of a project being depicted.
- Suggest and Explore Rather than Confirm - Sketches don't tell, they suggest.
- Ambiguity - Sketches are intentionally ambiguous, and much of their value derives from their being able to be interpreted in different ways

4. PERSKETCHTIVITY

PerSketchTivity is a web-based stylus and touch capable interface for students to learn and practice design sketching fundamentals. It incorporates the practices of traditional instructional pedagogy with artificial intelligence allowing students to practice a series of shapes with increasing complexity from simple and basic geometric shapes like Lines and Circles to complex three-dimensional perspective shapes; recognizing and analysing the sketches using sketch recognition techniques; and evaluating the sketches.

4.1 User Interface

4.1.1 Lessons and Challenges

The system has three main parts to it: Lessons, Challenges and Sketchbook. The lessons section has been organized into basic, perspective and primitive shapes. Figures 4.1, 4.2 and 4.3 show various lessons in the software. Currently, there are 8 lessons in overall that are active, each having its sub-lessons which have variations in angles, sub-types and visibility of vanishing point in perspective within the shape. For example, Lines lesson has 3 sub-lessons - Horizontal lines, Vertical lines and Diagonal lines. The following are the lessons and sub-lessons that are currently active: Lines - Horizontal, Vertical and Diagonal Lines; Curves - Horizontal Arcs, Vertical Arcs, Diagonal Arcs, S curves, Accelerated curves; Boxes - Squares and Rectangles; Circles - Three different levels of scaffolding; Planes - Planes in 2 point perspective with closer and farther vanishing points; Ellipses - Circles inscribed within square in 2 point perspective, has three different levels of scaffolding; 3D Boxes - Cubes and Cuboids; Cylinders

Challenges (figure 4.4) are designed to give the students an opportunity to en-

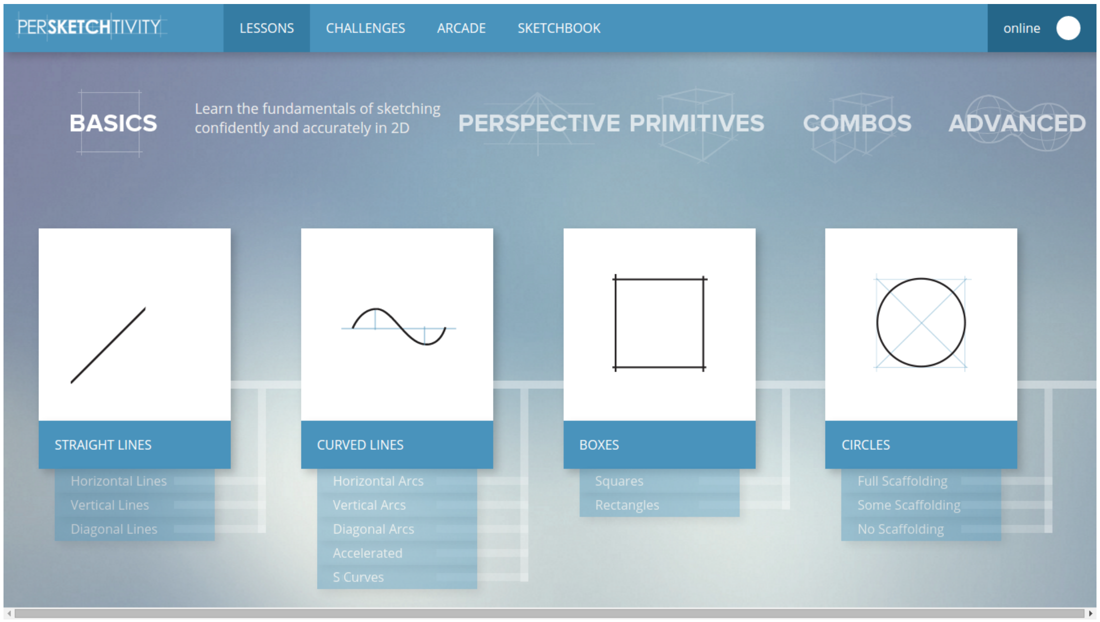

Figure 4.1: Basic shapes

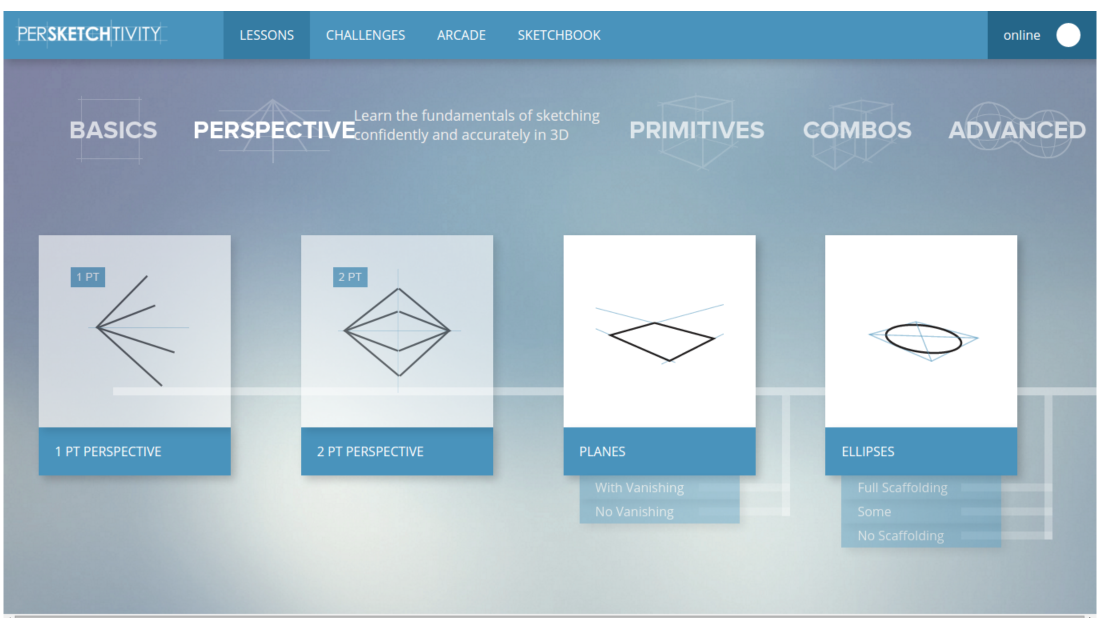

Figure 4.2: Perspective shapes

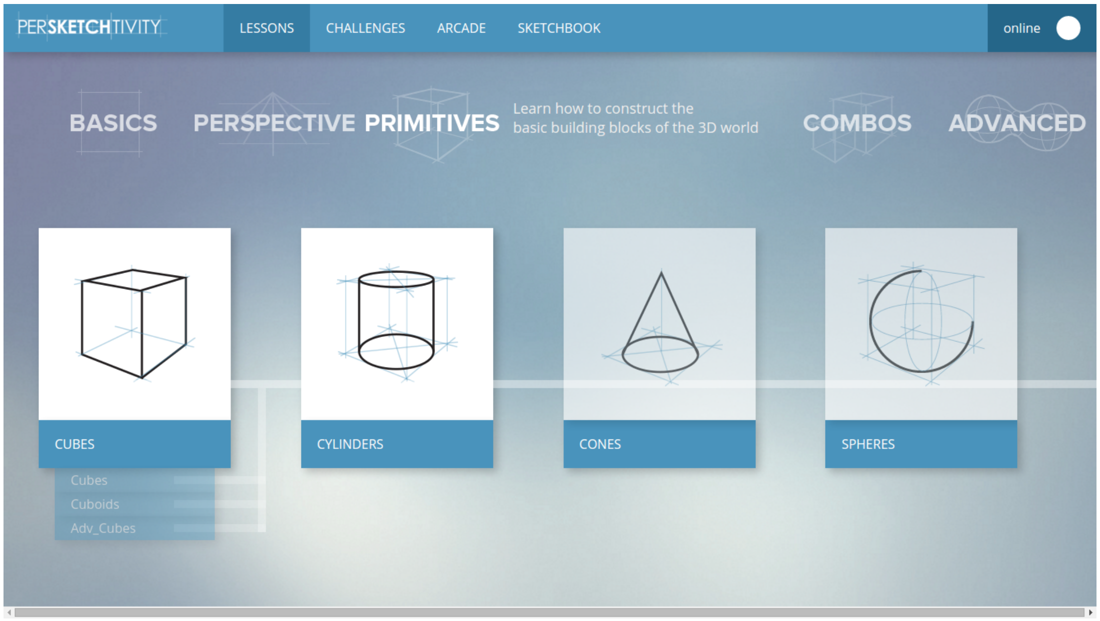

Figure 4.3: Primitive shapes

hance learning and improve their creativity, imagination, perspective skills much needed for design sketching. Sketchbook (figure 4.5) has a plain sketching area with different tools including pens and markers varying in various colors and thickness to choose from for free form drawing. Both challenges and sketchbook sketches can be saved by the user using the save button on the right bottom of the screen. It also has a delete button which erases all the strokes on the screen.

Each of the lessons has 8 exercises which are generated variations of the same problem by varying the parameters such as angle, length, size or perspective. These variants help students improve their muscle memory and hone their skill in drawing a particular shape. The screen displays visual cues to indicate what a student is expected to draw. For example, Lines lesson has two dots which student has to connect using a straight line. Also, students are given guidelines in the form of scaffolding to teach them proportions in drawing better. These scaffolding are removed slowly as

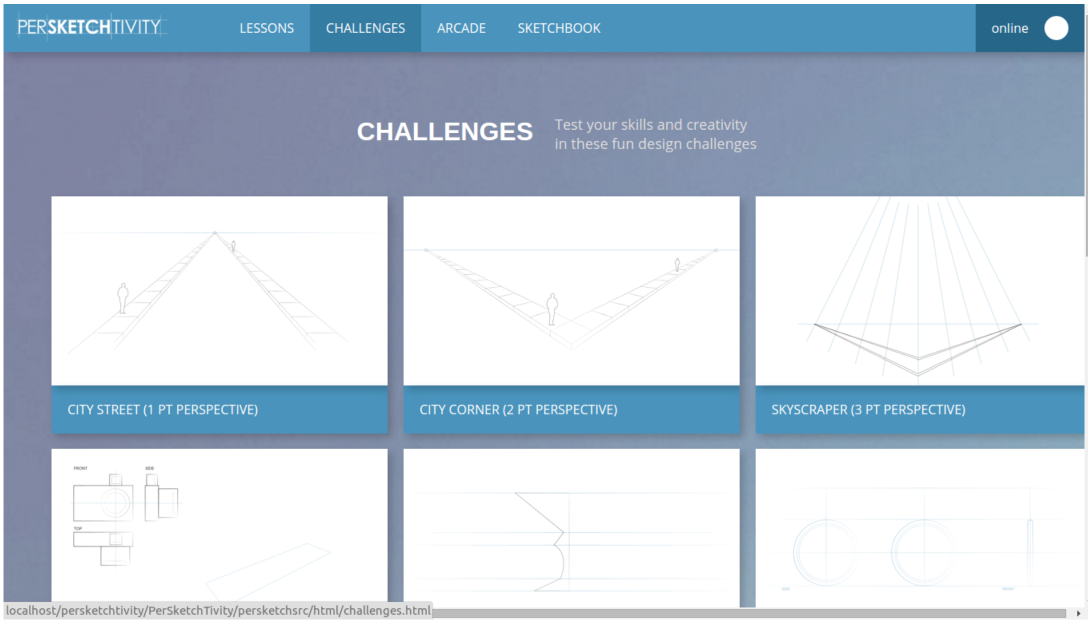

Figure 4.4: Challenges

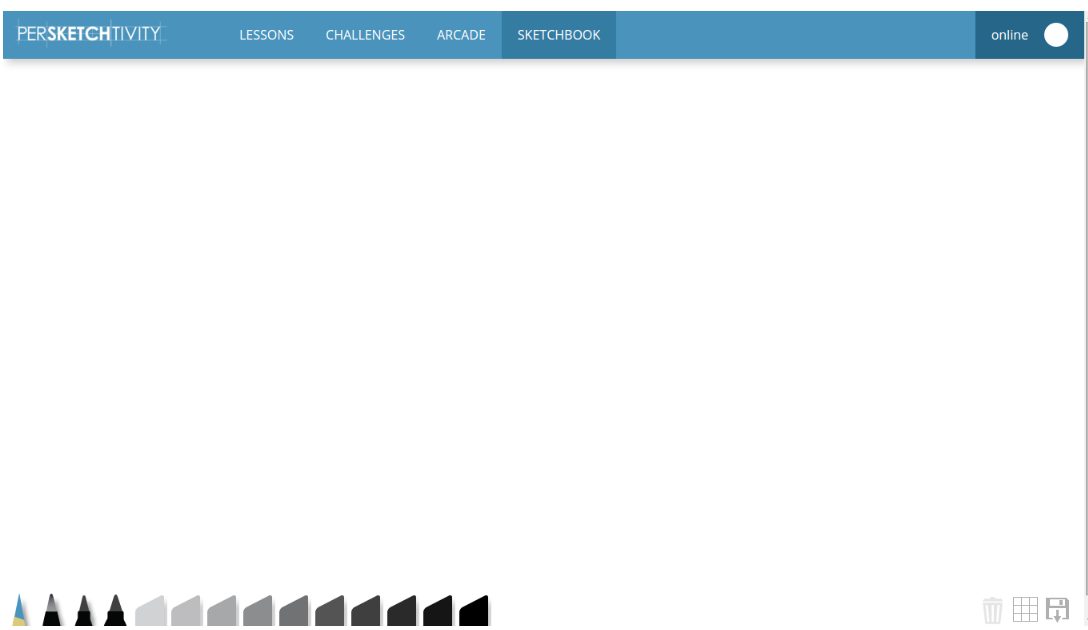

Figure 4.5: Sketchbook

one progresses to next sub-lessons. After the shape is recognized in an exercise, the system takes the student automatically to the next exercise. Students are evaluated and the results combining all of the exercises is displayed at the end of lesson.

4.1.2 Drawing Tools

The students can view any of the completed exercises in the current lesson by clicking on the review panel on the left hand side of the screen. This is a valuable tool for students to see how they are progressing through a lesson. Also, to help students while drawing there are three buttons provided on screen - UNDO, Skip and Next. UNDO button can be used to erase unintended strokes on the screen. Usage of this button is limited to once per exercise to restrict students from overusing the undo button to submit better sketches. Skip and Next button have similar functionality; while the Skip button is designed to be used to skip the current exercise in a lesson, next button progresses to next exercise when the students are done with drawing and the system failed to recognize it.

4.1.3 Architecture

The whole software is developed to work via Internet on mobile devices. The front-end is developed using HTML5, CSS3 and JavaScript. All the recognition and evaluation algorithms are executed by the browser using JavaScript. This makes the load on server much lesser and the duration needed to wait for the result considerably smaller. The system runs on Apache server. The server-side has been developed using PHP where all the functions relating to storing sketch data and user login are executed. The sketches and user details are stored using MySQL database.

The landing page for the website has a login form where the students are supposed to enter the credentials given to them to start using the system. The software stores each shape drawn by the user as a BLOB in the database. The system also stores the undone strokes to know the activity of users that led to final submission. Unfinished and unrecognized submissions that have been submitted by users are stored as well. The skipped lessons are not stored in the database.

4.2 Recognition system

4.2.1 Sketch Representation

Modern pen-based interfaces provide positional information along with current time in two dimensional coordinate system which is usually the window coordinates. The software generates a point as the pen or stylus moves over the input device. We record each of these point as the x-y coordinate along with its timestamp. The timestamp recorded is the epoch time which is time that has elapsed since 00:00:00 Coordinated Universal Time (UTC), Thursday, 1 January 1970 in milliseconds. A stroke is collection of time-ordered points which are between the pen-up and the pen-down events. A sketch is comprised of one or more strokes and it is defined as a shape when it satisfies geometric constraints and is recognized by our software.

4.2.2 Preprocessing

Before passing strokes to our recognition system, it is necessary to preprocess the strokes. Sezgin and Davis [79] and [67] state that the input strokes will contain noisy and inaccurate samples which is caused by spatial and temporal quantization of the input by the hardware. Spatial digital noise comes from conversion of ink positions to screen coordinates. The difference in the sampling rates of the operating system and the tablet causes temporal quantization errors. This motivates us to eliminate and decrease the problems caused by such noise and we resample points and time. This also helps subsequent steps in the process easier.

4.2.2.1 Resample points

The number of points sampled by the system depends on the speed of the pen movement. The number of points will be lesser when the speed is more and more when the speed is less. We use an algorithm similar to $1 recognizer [99] with one difference. Instead of the target stroke having a fixed number of points, we resample in such a way that the points are equidistant from each other. We choose this threshold to be 2. We choose this technique because it is simple to implement.

4.2.2.2 Resample time

It was noticed that the time stamps recorded for the points of the stroke that are drawn rally fast are not accurate. The time stamp remains the same for a lot of consecutive points. This causes problem while calculating speed as mentioned by Sezgin and Davis [79]. To solve this issue we resample time as well. We interpolate the time based on the distance between points for those that have same time stamp by using the time stamp of first point in the sequence of points having same time and the first point after this sequence.

4.2.3 Merging and Segmenting

To make recognition easier by using geometric constraints we merge and segment the strokes based on the requirements of our recognition algorithm. For squares, rectangles, cubes and cuboids we segments the strokes to obtain individual edges of the shape. For circles and ellipses we merge the strokes which satisfy the following conditions: (i) The strokes are consecutive. The difference in the time stamps between the last point of the first stroke and the first point of the next stroke are less than a threshold. (ii) The strokes are close to each other. The distance between one of the end points of one stroke is close to one of the end points of the another stroke.

4.2.4 Overtracing and Overdrawing

Overtracing is the technique of drawing above the already drawn stroke. This is a natural way of drawing closed shapes like circles and ellipses. In these specific case of circle, it is usually observed that the end of circle overlaps with the beginning of the stroke. To facilitate this while keeping in mind our recognition and evaluation components of the system, we allow users to overtrace until it is within some threshold and ignore this for our future processing of the strokes.

For straight lines connecting two given end points, the correct technique of drawing in design sketching is to draw beyond the end points to make the line quality better. This technique is called overdrawing. In our system we allow students to draw this way and the overdrawn stroke will be considered in further steps of recognition and evaluation.

4.2.5 Recognition

To achieve recognition, we rely on the state-of-art techniques of sketch recognition. The research in this field can be categorized into three sub-fields: geometric recognition, gesture-based recognition and vision-based recognition. A lot of work has been done in each of these categories. We use geometric recognition to identify the shape of the sketches drawn by students. The user interface lets a student draw strokes until the shape is recognized. Each time a stroke is drawn the recognition system takes that in, combines with the previous drawn strokes and checks to see if the geometric constraints for the shape are satisfied. Once recognition is completed, the strokes are sent to the next stage for evaluation which is explained in the next chapter.

5. GRADING RUBRIC

One of the essential elements for an Intelligent tutoring system is to provide immediate feedback to the students which help in identifying the errors and guide them towards better understanding of the concepts [5]. Thus, assessing sketches and giving feedback is a very significant aspect of PerSktchTivity. In this chapter, we explain different rubrics we use to assess the student sketches.

5.1 Pre-evaluation Processing

Before sending the sketches to the evaluation system, we preprocess the segmented and merged strokes to ignore the hooks at the start and end of the stroke that are caused when drawing fast. We remove 5% of the stroke at either ends for further analysis. To not remove number of points from the stroke to an extent that we lose significant information about it, we restrict the number of points removed to be a maximum of 5 points.

5.2 Categories for Grading Rubric

To design an evaluation system, the first step is to get useful features from the sketch which determine the quality of sketch. We have divided our evaluation features into three categories as given below. They are in the increasing order of hierarchy of the components of sketch they use in evaluating.

- Visual - The features in this category are dependent on how the final sketch looks. We consider only the x-y coordinates of the points on the sketch for all features in this category.

- Technique - These are set of features that are controlled by the manner of movement of pen over the screen. The features in this category include the time stamps of the points as well along with x-y coordinates of the ink on the screen.

- Planning - Set of features which demonstrate the high level planning in drawing the whole shape. These features tell us the way the strokes of the shape are drawn.

The set of features that have been used in this thesis for evaluating sketches have come from the following listed sources -

- Characteristics of sketches (mentioned in section 3.3)
- Features used in sketch recognition
- Interviewing experts
- Features from motor-control studies ([26], [2], [64])
- That we believe are important to identify the expertise level of the user

5.3 Motivations for Grading Rubric

This section briefly explains the motivation for choosing the initial set of features. Detailed explanation of the features is given in the following section

5.3.1 Visual

Design sketches are drawn to represent ideas in visual forms and share with others. An important aspect in this is to be able to visually depict the idea that is conceived in one's mind and being able to express it on paper. Inaccurate design sketches becomes worthless. Thus the visual accuracy, realism and neatness of the sketches become important. One of the important reasons why CAD became popular was because one could draw accurate and precise designs using it. We consider two different measures to evaluate sketches in PerSketchTivity based on this desired property of design sketches as explained below.

- Accuracy - Accuracy refers to the degree of conformity and correctness of the shape compared to the expected shape. In other words, this represents how close the sketch is to the expected shape.
- Smoothness - Smoothness refers to a state of consistency in the sketch. A wavy and jittery sketch is not neat and elegant. This is another way of seeing the precision of the sketch.

Both accuracy and smoothness are needed in a sketch for it to convey the ideas of the person drawing it. An accurate sketch may not be smooth and vice-versa. Figures in 5.1 illustrate this. While the former is accurate, it is not smooth and the later is smoother but its accuracy is bad.

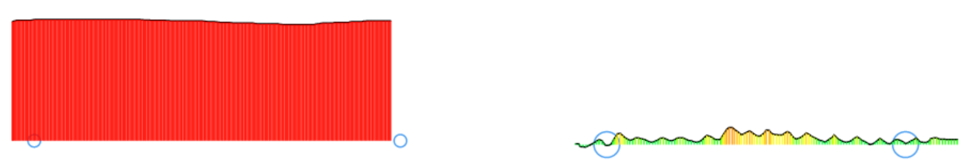

Figure 5.1: Left: Good smoothness, bad accuracy, Right: Good accuracy, bad smoothness

5.3.2 Technique

The motivation behind this category of features is to see how an user moves hand during the process of sketching. Sketching is an acquired skill and the experts, who have a lot of experience in sketching, have better muscle memory and hence the way the experts draw will be significantly different from the novices.

- Speed - A designer draws multiple sketches in the early stages of the process to be able to decide what works and what does not. Hence, it becomes important for him/her to be quick enough in generating the sketches and put his ideas on paper in visual form. This is the feature that has been derived from design sketching practices. This is also one of the qualities of sketches mentioned in the list given in the Chapter three. This property can help us distinguish between the users.

- Speed Fluidity - There has been a lot of research in the past for many decades on the model of human movement in human-computer interaction [26, 27]. Our motivation for this particular feature is from the research on the minimum jerk's law and trajectory based fitts' law [2]. The speed of the motion is maximum in the middle of the path of the movement and decreases as you move away from this. We analyse how well this law works in the act of sketching and how much better do experts flare in this.

- Accuracy vs speed - There has also been research in applying fitts' law to find many other interesting relationships in hand movement. Meyer provides details of how speed and accuracy have a trade-offs in aimed movements [64]. Similar trade-offs can be applicable to trajectory based movements like using a stylus on tablets as well. This motivates us to further analyse the relationship between these features and see how they differ in experts and novice users.

5.3.3 Planning

An experienced designer who has more knowledge about the concepts of design sketching and the techniques will plan better initially before starting to draw and would have an idea of what he/she is going to do. He/She has has high enough experience to know which strokes to drawn next and drawing in what order will make the sketch look good. These set of features are basically developed from the motivation from expert knowledge and referring design sketching textbooks.

- Overdrawing - A recommended practice is that the lines be drawn by drawing beyond the end points to have a better quality line between the end points. An inexperienced person may not know this or know the importance of this technique and may fail to follow it.

- Breaking or Coupling Strokes - This feature checks how the user tends to draw a shape on being given guides. Let us take an example to understand this better. A user can drawn a square using a single stroke or multiple strokes. With this feature we want to examine this behavior or experts and novices

- Stroke Order - When drawing a complex shape, one tends to use multiple strokes. There are exponential permutations of drawing the shape using multiple strokes. This feature helps us investigate the order of strokes that experts and novices use.

- Stroke Direction - A stroke can be drawn from left to right or right to left, bottom to top or top to bottom for line, clockwise and anti-clockwise for circles and ellipses.

5.4 Features used

5.4.1 Accuracy

To measure accuracy of a shape with respect to the ideal expected shape based on the visual guidelines given, called the reference shape, we use motivation from vision based sketch recognition algorithms [99, 55]. For a given sketch, we find the distance between each of the points on the sketch and its corresponding point on the reference shape to calculate the deviation at that point. The measures used for measuring accuracy are

- Maximum deviation - The maximum of all deviations
- Average deviation - It is calculated by adding the deviations at every point and dividing it by the total number of points.
- Hausdorff similarity - Hausdorff distance is one of the popular methods to compare how similar two sketches are [55]. Hausdorff distance between two set of points A and B is given by the equation

$$H(A, B) = \max(h(A, B), H(B, A)) \tag{5.1}$$

where

$$h(A, B) = \max_{a \in A}(\min_{b \in B \|a-b\|}) \tag{5.2}$$

We use this distance and get the similarity measure for the sketch using the equation given below [1]. When Hausdorff distance is zero, the similarity is 100%.

$$similarity(A, B) = 1 - (H(A, B).\frac{\sqrt{2}}{300})^{\frac{1}{1.4}} \tag{5.3}$$

[1]http://goo.gl/52OsBi

5.4.1.1 Lines

For lines, deviation at each point is calculated by finding a point on reference shape that is closest to it. This is same as the drawing a perpendicular line to the reference shape from the point on the sketch. The figure 5.2 illustrates this.

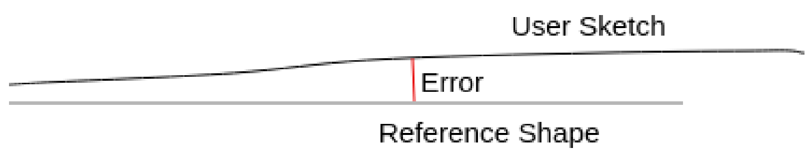

Figure 5.2: Deviation of horizontal line at one point

5.4.1.2 Circles and Ellipses

The distance between a point on the sketch and the point on the reference shape which is at the same radial angle to the center is the deviation of a point on circle or ellipse. We first connect the point to the center and then find the point of intersection of that line on the reference shape to get the deviation of a point.

5.4.1.3 Square and Rectangles

Squares and rectangles are made of four sides. The evaluation system is given 4 strokes that make up the sketch of the square the user drew. They are merged and segmented before recognition phase. We find the corresponding line on the reference

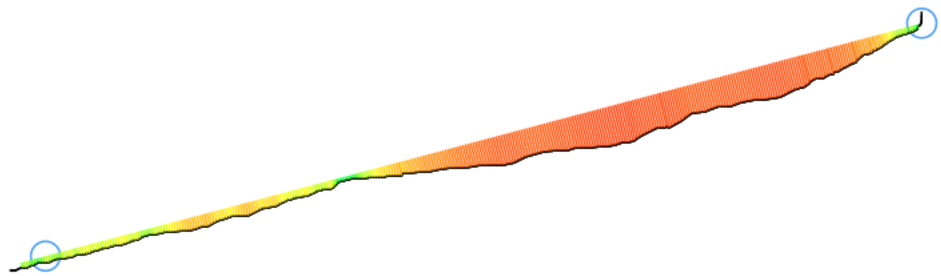

Figure 5.3: Deviation of a diagonal line

shape which is closest the user line and implement the same algorithm that we used for lines to get deviation at each point on these lines.

5.4.1.4 Cubes and Cuboids

Cubes and Cuboids are made of 12 edges. For this also, like squares and rectangles, the user strokes are merged/segmented to form 12 separate lines representing the edges in the preprocessing stage and then the corresponding edges of the reference shape which is closest to the user stroke is identified. After this a similar method used for rectangles and squares is used to get deviation for each point on the drawn cuboid.

5.4.2 Smoothness

Texture of the sketch defines the smoothness. It is a way to represent the precision of the sketch. Smoothness helps in measuring the closeness of the sketch to the desired shape. It is defined as the measure of deviation of the stroke from the ideal

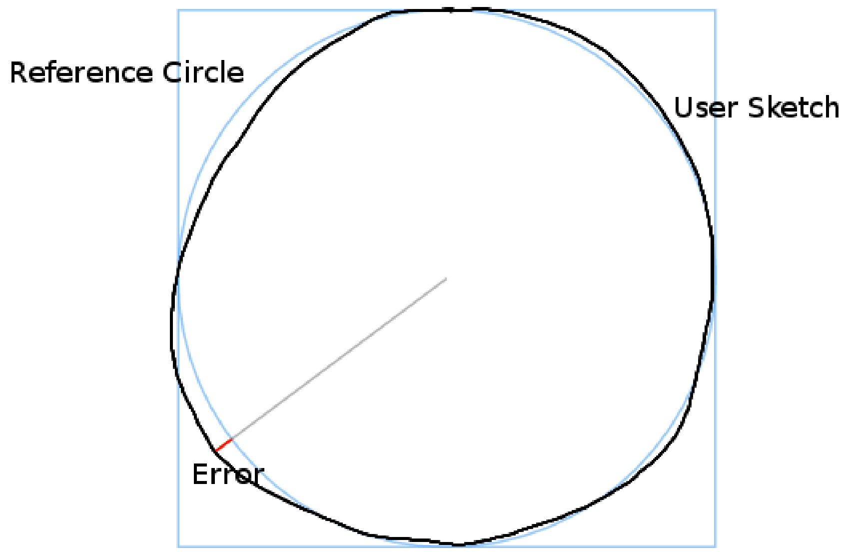

Figure 5.4: Deviation of circle at one point

shape that the user was intending to draw. A smooth stroke is drawn steadily without jolts during the action of sketching. Mathematically, the strokes which are differentiable at least once are smooth. A good quality sketch should have no unwanted kinks or cusps.

For calculating smoothness, we take motivation from three of the Rubine [77] features representing the curvature and sharpness. While the first two features denote the curvature of the stroke, third feature distinguishes smooth strokes and those with sharp angles. The absolute angle of a 'V 'curve and 'U'curve can be same, but the sharpness feature helps in distinguishing between them. Angle at a point p is calculated as shown in figure 5.8 using the equation 5.4

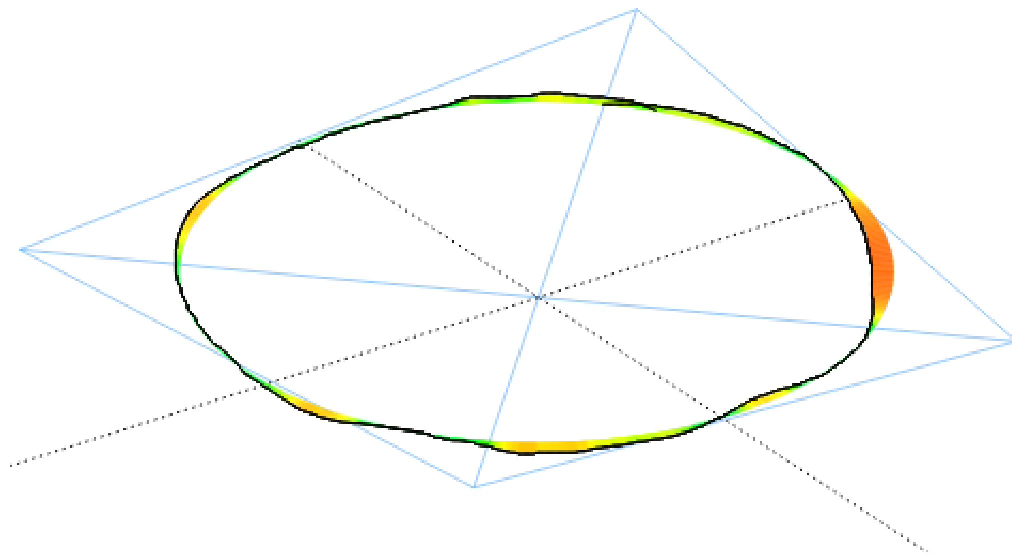

Figure 5.5: Deviation of ellipse

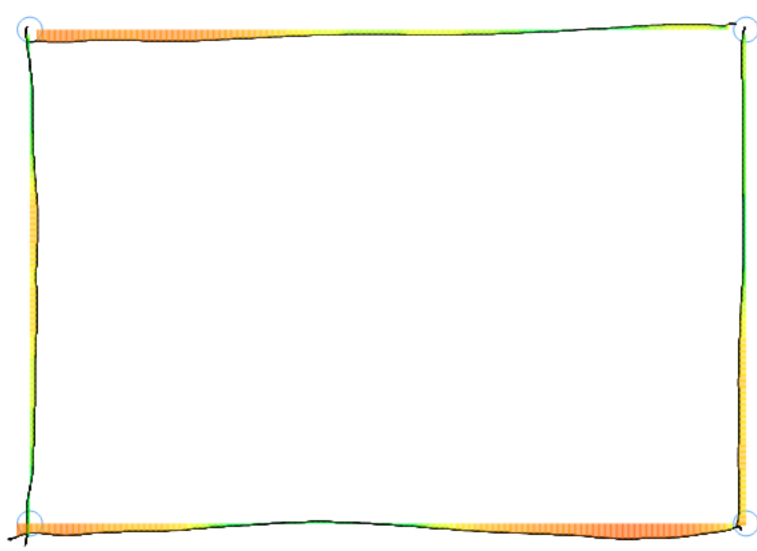

Figure 5.6: Deviation of rectangles

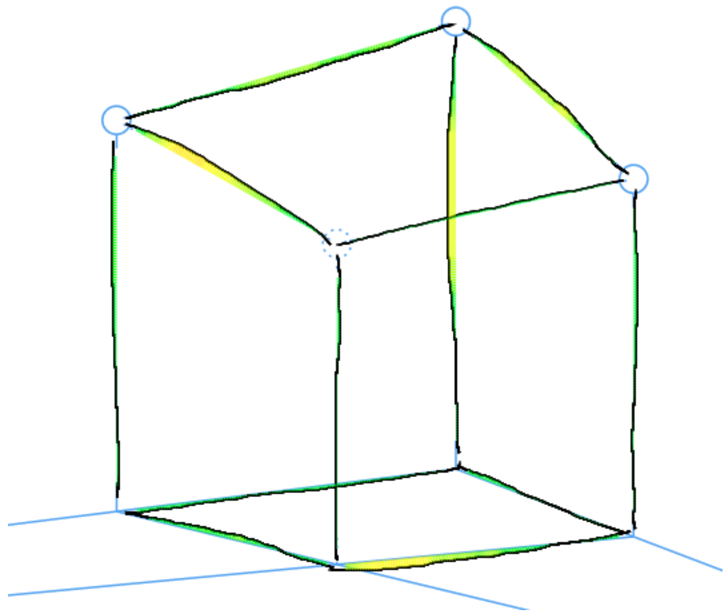

Figure 5.7: Deviation of cubes

$$\theta_p = \arctan \frac{\Delta x_p \Delta y_{p-1} - \Delta x_{p-1} \Delta y_p}{\Delta x_p \Delta x_{p-1} - \Delta y_p \Delta y_{p-1}} \tag{5.4}$$

The following features are measured to know the smoothness of the sketch:

- Maximum Absolute Angle - maximum of absolute angle at every point
- Average Angle - average of angle at every point at every point

$$AverageAngle = \frac{\sum_{p=1}^{p-2} \theta_p}{p-2} \tag{5.5}$$

- Average Absolute Angle - average of absolute angle at every point

$$AverageAbsoluteAngle = \frac{\sum_{p=1}^{p-2} |\theta_p|}{p-2} \tag{5.6}$$

- Average Square Angle - average of square of angle at every point

$$AverageSquareAngle = \frac{\sum_{p=1}^{p-2} \theta_p^2}{p-2} \tag{5.7}$$

5.4.2.1 Lines

For Lines, at each point the angle between two lines - one joining the point and previous point and another joining the point and the next point is calculated. The figure 5.8 demonstrates how angle is calculated between points p1,p2 and p3.

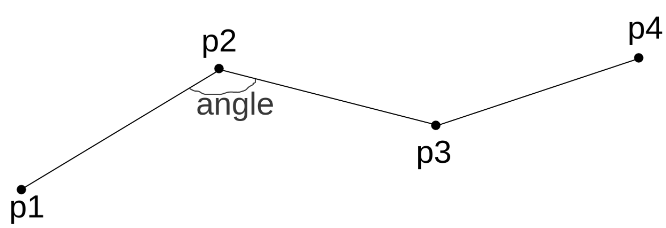

Figure 5.8: Smoothness calculation of lines

5.4.2.2 Circles

To get the smoothness of Circles, it is not possible to use what has been used for lines because of the curvature of the circle. The angle between two adjacent lines joining consecutive points in a circle will always be at an angle and this is dependent on the radius of the circle. To simplify the calculations of circle, the reference circle is stretched in such a way that it forms a straight line. The points on the user sketch are also stretched such that the distance between points on the original user sketch and the corresponding point on the original reference shape which is at same radial angle as user point is same after stretching the shape as well. Figure 5.9 shows this concept for better understanding.

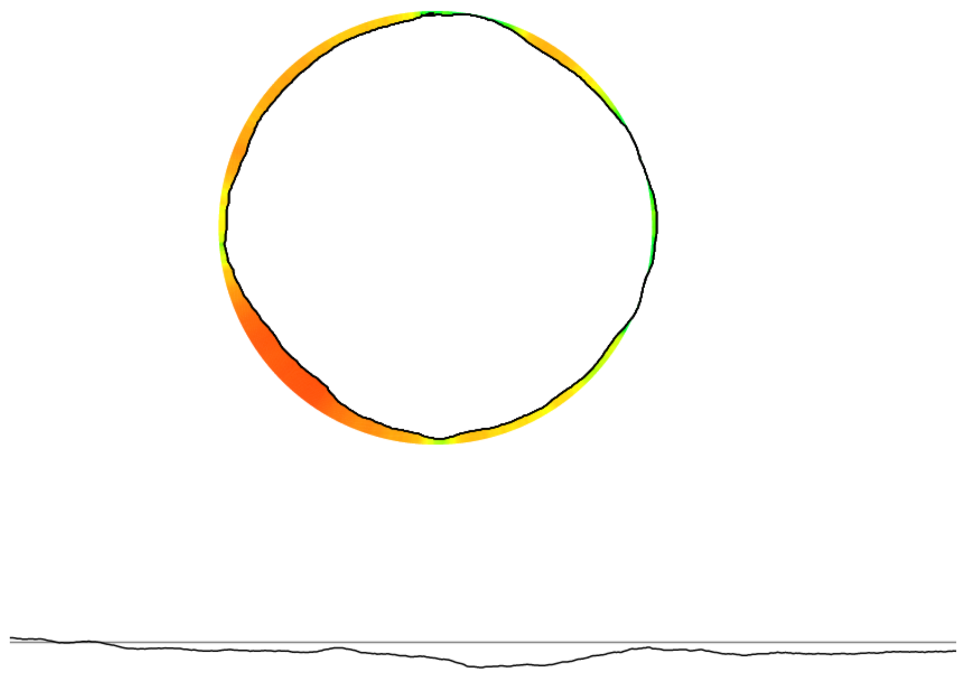

Figure 5.9: Smoothness calculation of circles

5.4.2.3 Ellipses

We implement an algorithm similar to that of circle for the ellipses as well. The ellipse reference shape and the sketch is stretched to get a straight line from the reference shape.

5.4.2.4 Squares and Rectangles

Smoothness of each of the four edges is measured separately to come up with a single value for each of the features mentioned in the list above.

5.4.2.5 Cubes and Cuboids

Each of the twelve edges of cuboids and cubes are taken individually and their smoothness is measured to get the smoothness features of the whole shape.

5.4.3 Speed

Speed quantifies how fast the strokes in the sketch was drawn by the user. It measures the rate at which the user moves the pen over the screen of the tablet. It is calculated by using the following formula for speed

$$speed = \frac{distance}{time} \tag{5.8}$$

At each point the distance covered from the previous point is divided by the difference in their time stamps to get the speed. The features that have been used to measure speed are:

- Maximum speed
- Minimum speed
- Average of speeds between consecutive points
- Average speed given by

$$\frac{TotalPathLength}{TotalTimeTaken} \tag{5.9}$$

For all the shapes these features are calculated in the same way. For complex shapes which have multiple strokes, the speed of each stroke is calculated to give a value for the whole shape together. Average speed is calculated by using the total path length traversed by each of the strokes divided by the time taken to finish the whole shape, that is start time of the first stroke and the end time for the last stroke.

5.4.4 Speed Fluidity

Using this feature we can measure how the speed changes during the drawing of the stroke. A lot of research has been done in the area of movement of hand which we try to implement on the sketches in PerSketchTivity. This is represented by the graph given in figure 5.10

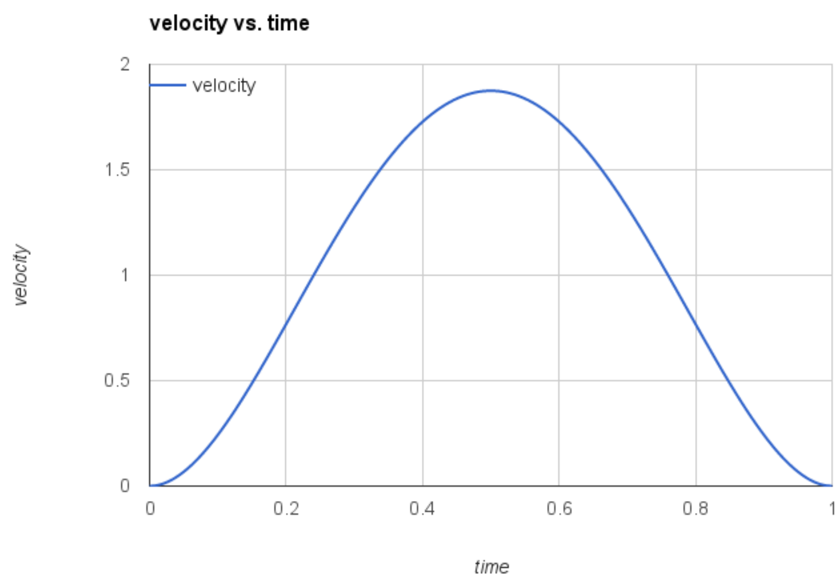

Figure 5.10: Velocity vs time graph while moving hand in space towards a target

The following features have been used to measure it.

- Speed fluidity ratio_1 - This is the ratio of speed between the first point and the point at 3/4th of the distance of the stroke and the speed between the point at 3/4th distance of the stroke and the last point. Since the speed decreases in the end, the ratio is expected to be more than 1

- Speed fluidity ratio_2 - This is the ratio of speed between the points at 1/4th distance and 3/4th distance of the stroke to the speed between the points at 3/4th distance and the last point.

5.4.5 Speed Vs Accuracy

There has been research on the trade-off between accuracy and speed in the motor skills while moving hand to a aimed target [64, 98, 105, 50]. To see if this kind of relationship exists in the act of sketching as well, we add the following features to our set

- Ratio of average deviation to average speed
- Ratio of maximum deviation to maximum speed

5.4.6 Stroke Order

This feature lets us check in what order the strokes of a complex shape which can have multiple strokes are drawn. This does not mean anything for lines, circles and ellipses. For the other two shapes, we describe the algorithm used briefly below.

5.4.6.1 Squares and Rectangles

We assign a binary score of 0 or 1 to the shapes based on the drawing of parallel lines together. We give the sketches in which the parallel lines have been drawn one after the other a score of 1, that is drawing the top and the bottom horizontal side of the shape together and left and right vertical side of the square together, and the rest of the sketches a score of 0.

5.4.6.2 Cubes and Cuboids

A binary score of 0 or 1 is given the cubes and cuboids depending on the order of drawing the top face, bottom face and the edges. We give a score 1 when all the sides of one face or the vertical edges are drawn together one after the other before proceeding to other face or the vertical edges

5.4.7 Stroke Direction

This feature helps us measuring the direction of each of the stroke in the shape and give the sketch an overall value. The possible directions for a straight line are top-to-bottom, bottom-to-top, right-to-left and left-to-right. Two possible directions for circles and ellipses are clockwise and counter clockwise

5.4.7.1 Lines

For horizontal lines, strokes drawn from left to right are given a score of 1 and the ones from right to left are given -1. Similarly, for vertical lines we give a score of 1 and -1 for lines being drawn from top-to-bottom and bottom-to-top, respectively. In case of diagonal likes, we treat them like horizontal or vertical lines depending on the angle they make with the X-axis. We use the grading score similar to the closer of the two possibilities - horizontal and vertical.

5.4.7.2 Circles and Ellipses

For circles and ellipses, we assign a score of +1 and -1 for clockwise and anti-clockwise respectively. To find the direction we consider 4 points on the stroke which are equidistant from each other, which will be first point, last point and two points in between. Then we use the following equation to get the direction.

5.4.7.3 Squares and Rectangles

We give a score ranging from 0 to 2 depending on the direction of each of the component strokes. When the strokes that are parallel are drawn in the same direction, the score is incremented by 1.

5.4.7.4 Cubes and Cuboids

Scores between 0 and 5 is given based on direction of the component strokes in the sketch. Drawing parallel sides of the cuboid with strokes in same direction moves the stroke towards 4.

5.4.8 Stroke Coupling and Breaking

This feature helps us measure the way the strokes are coupled and broken to get the final recognizable sketches. It is speculated that the quality sketches will break the shape into components and try to draw them individually to get higher realism and better quality. We number of strokes in the final shape to determine this. We do not consider the stray strokes and the strokes which were removed using UNDO button.

5.4.9 Overdrawing

The designers draw the lines beyond the desired end points to get a better quality of line between the end points. To check this feature we added a feature which is the ratio of length of the stroke to the total length of the whole shape. For complex shapes, the length of the strokes is added to get the path length of the sketch.

6. EVALUATION

It is necessary to understand how well a computer can distinguish between a novice and an expert by using the set of features we believe are important for a good sketch to design an assessment system for sketches. Analysis which of the the features works better in distinguishing the two sets of users as well is helpful to reach the goal of automatic grading system. This section describes the experiments conducted to find this out and discusses the results obtained from the experiments.

6.1 Data Collection

All the users were given separate credentials to login and draw sketches in each of the lessons using our software PerSketchTivity. To maintain consistency, they were all asked to use Wacom Cintiq tablet. All the users were told to go through five of the lessons - Lines, Circles, Rectangles, Ellipses and Cuboids.

Novice data - We used data collected at Georgia Tech from 20 students enrolled in the ME/CE 1770, Introduction to Engineering Graphics and Visualization. The class predominantly has freshmen students.

Expert data - We collected data from 4 different experts having expertise in the area of design sketching. They all had good level of comfort in using Wacom Cintiq tablets. One of them is an instructor for the course, two other lecturers and professors in the Industrial Design Department at Georgia Tech, and another Master's graduate student who has been TA for the course.

For this study, we use only the sketches that have been recognized by our recognition system, incomplete sketches that the system failed to recognize have not been included. A total of 2627 sketches collected from both experts and students was used for our analysis. Table 6.1 shows the number of shapes for each of the shapes for both expert and novice.

Table 6.1: Number of sketches

Shapes	**#Student sketches**	**#Expert sketches**
Lines	343	62
Circles	952	103
Rectangles	231	61
Ellipses	548	61
Cuboids	224	42

Once the data was collected, we used our evaluation system to get all the feature values that we discussed in the previous chapter and other features that we believed are important and stored them in the database.

6.2 Analysis and Results

6.2.1 Correlation Between Features

To see the relationship between different parameters while drawing, we perform a correlation between the different features discussed above. The analysis showed that there is moderate correlation between some of the features. The pair of features which has this kind of relationship were: (i)speed and smoothness - positive correlation, (ii)accuracy and size of shape - negative correlation. So we added some more features to check for these relationships in the data. The table 6.2 shows the correlation for all the shapes

6.2.2 Experts Vs Students

The first step in developing a grading system for this thesis is to check which of the features are helpful in making a good quality sketch. For this analysis, we check to see which of the features work well in distinguishing students who are novices

Table 6.2: Correlation between features

Pair of features	Lines	Circles	Rectangles	Ellipses	Cuboids
Average absolute angle and average speed	-0.25	-0.75	-0.65	-0.64	-0.58
Average square angle and average speed	-0.41	-0.67	-0.47	-0.56	-0.23
Average deviation and size of shape	0.10	0.544	0.43	0.10	0.47

and the experts who have knowledge as well as experience in sketching. The goal of teaching sketching to students is to make them as skillful as the experts and make them capable of drawing expert quality sketches. There are multiple ways of doing this using univariate analysis. We use statistical analysis on each of the features and subset selection to achieve this. Welchs t-test is a statistical test to determine if two sets are significantly different from each other. Subset selection or feature selection is the process of selecting relevant features that can be used in constructing a model.

6.2.2.1 Statistical Analysis

There are two tests we use to see if the features are significantly different from each other. We performed t-test for each of the features on expert and student data. Welch's t-test is used to see whether means of two sets of data is significantly different from each other. We use t-test to understand if these features can be used to distinguish the users and which of these features might be more useful in disambiguating the users. The results of running t-test for lines, circles, rectangles and cubes is given in table 6.3, table 6.4, table 6.5, table 6.6 and table 6.7, respectively. Any feature having p-value below alpha=0.05 is considered significant and have been marked in bold in the tables.

Table 6.3: T-test results for lines

Features	t-value	p-value
Average deviation	4.636166	**0.000004**
Maximum deviation	4.732773	**0.000003**
Hausdorff similarity	-5.251743	**0.000000**
Maximum speed	-6.116622	**0.000000**
Minimum speed	-3.110715	**0.002095**
Average speed 1	-6.290789	**0.000000**
Average speed 2	-4.553132	**0.000011**
Maximum angle	3.989594	**0.000107**
Average angle	0.613916	0.539506
Average absolute angle	12.595520	**0.000000**
Average square angle	3.558509	**0.000405**
Ratio of middle 50% to last 25% of stroke	-4.109492	**0.000078**
Ratio of first 75% to last 25% of stroke	-4.776301	**0.000006**
Stroke Direction	3.068742	**0.002677**
Ratio of path length to reference length	-2.593129	**0.010119**
Ratio of average absolute angle to average speed	4.372970	**0.000015**
Ratio of average square angle to average speed	1.654864	0.098576
Ratio of average deviation to average speed	2.838122	**0.004721**
Ratio of maximum deviation to maximum speed	3.584605	**0.000370**
Ratio of average deviation to size of shape	2.725056	**0.006653**

Table 6.4: T-test results for circles

Features	t-value	p-value
Average deviation	1.142722	0.255167
Maximum deviation	0.295209	0.768407
Hausdorff similarity	0.206242	0.836916
Maximum speed	-1.951687	**0.052050**
Minimum speed	-5.722706	**0.000000**
Average speed 1	-3.688476	**0.000282**
Average speed 2	-6.477410	**0.000000**
Maximum angle	-3.354055	**0.001066**
Average angle	-2.612191	**0.010087**
Average absolute angle	6.163304	**0.000000**
Average square angle	5.724333	**0.000000**
Ratio of middle 50% to last 25% of stroke	0.875700	0.381595
Ratio of first 75% to last 25% of stroke	0.782068	0.434824
Number of strokes	-0.167188	0.867487
Stroke Direction	1.372872	0.172268
Ratio of path length to reference length	-0.104416	0.916945
Ratio of average absolute angle to average speed	9.239753	**0.000000**
Ratio of average square angle to average speed	9.285740	**0.000000**
Ratio of average deviation to average speed	9.095383	**0.000000**
Ratio of maximum deviation to maximum speed	7.749994	**0.000000**
Ratio of average deviation to size of shape	3.142940	0.002001

Table 6.5: T-test results for rectangles

Features	t-value	p-value
Average deviation	5.723932	**0.000000**
Maximum deviation	5.131870	**0.000001**
Hausdorff similarity	-3.363090	**0.001023**
Maximum speed	-1.098152	0.276525
Minimum speed	-8.323102	**0.000000**
Average speed 1	-6.540074	**0.000000**
Average speed 2	3.793007	**0.000245**
Maximum angle	6.905698	**0.000000**
Average angle	0.733288	0.464098
Average absolute angle	14.177324	**0.000000**
Average square angle	11.858442	**0.000000**
Ratio of middle 50% to last 25% of stroke	-2.389580	**0.019862**
Ratio of first 75% to last 25% of stroke	-2.408773	**0.018794**
Number of strokes	-11.423449	**0.000000**
Stroke order	-13.062777	**0.000000**
Stroke Direction	-1.000000	0.321327
Ratio of path length to reference length	-4.482404	**0.000025**
Ratio of average absolute angle to average speed	8.929782	**0.000000**
Ratio of average square angle to average speed	7.851298	**0.000000**
Ratio of average deviation to average speed	11.185959	**0.000000**
Ratio of maximum deviation to maximum speed	8.736495	**0.000000**
Ratio of average deviation to size of shape	10.777805	**0.000000**

Table 6.6: T-test results for ellipses

Features	t-value	p-value
Average deviation	-1.223841	0.221699
Maximum deviation	-0.260040	0.795042
Hausdorff similarity	2.696890	**0.008513**
Maximum speed	-6.003373	**0.000000**
Minimum speed	-8.421616	**0.000000**
Average speed 1	-14.527838	**0.000000**
Average speed 2	-14.417200	**0.000000**
Maximum angle	-0.841677	0.402854
Average angle	0.237366	0.812988
Average absolute angle	27.947944	**0.000000**
Average square angle	19.619505	**0.000000**
Ratio of middle 50% to last 25% of stroke	3.503697	**0.000724**
Ratio of first 75% to last 25% of stroke	6.092394	**0.000000**
Number of strokes	2.846700	**0.004583**
Stroke Direction	2.202383	**0.030775**
Ratio of path length to reference length	-6.868849	**0.000000**
Ratio of average absolute angle to average speed	22.091111	**0.000000**
Ratio of average square angle to average speed	18.880658	**0.000000**
Ratio of average deviation to average speed	7.981763	**0.000000**
Ratio of maximum deviation to maximum speed	10.041888	**0.000000**
Ratio of average deviation to size of shape	1.628801	0.103888

Table 6.7: T-test results for cuboids

Features	t-value	p-value
Average deviation	-0.472613	0.638938
Maximum deviation	-0.881524	0.383147
Hausdorff similarity	0.066592	0.947102
Maximum speed	-6.385312	**0.000000**
Minimum speed	-3.679507	**0.000660**
Average speed 1	-18.014949	**0.000000**
Average speed 2	-2.147321	**0.035358**
Maximum angle	-2.993699	**0.004452**
Average angle	-1.848185	0.070930
Average absolute angle	17.716607	**0.000000**
Average square angle	9.527313	**0.000000**
Ratio of middle 50% to last 25% of stroke	-3.503003	**0.001062**
Ratio of first 75% to last 25% of stroke	-1.131489	0.263953
Number of strokes	-3.740387	**0.000225**
Stroke order	-0.907470	0.368091
Stroke Direction	-5.443589	**0.000000**
Ratio of path length to reference length	-5.157017	**0.000000**
Ratio of average absolute angle to average speed	13.034510	**0.000000**
Ratio of average square angle to average speed	10.248731	**0.000000**
Ratio of average deviation to average speed	15.567807	**0.000000**
Ratio of maximum deviation to maximum speed	2.642749	**0.011185**
Ratio of average deviation to size of shape	0.978338	0.333450

6.2.2.2 Subset selection

For T-test a number of assumptions on the data being used should be true. Also, one of the limitations of the t-test is that it only tells if the means of the two sets are significantly different from each other. Lower p-value does not guarantee that the individual values will be significantly different from each other as well. So, we decided to do subset selection on all the features for all shapes to verify that the set of features we have got from T-test are the significant ones for our study. Subset selection is a method used to select a group of features from the available features so that this subset represents the the data well [8]. We used Weka [29] which is an open source software package written in Java containing algorithms machine learning and data mining. Table 6.8 shows the features that were selected using this method. The cells marked with ✗ are the ones which are not significantly different using t-test. For the final set of features we combine the sets of features found from the t-test and subset selection method.

6.3 Discussion

The statistical analysis we performed using t-test shows that most of the features for Lines and Rectangles are significant. One interesting observation from the results is that the accuracy of the experts and students for shapes including circles, ellipses and cubes does not differ significantly. This is congruent with the whole theory of design sketching. Design sketches are used to quickly get as many ideas as possible on the paper to share it with others, generate further ideas. Hence, experts do not give attention to the accuracy and concentrate on other things important in sketches.

The hausdorff similarity in case of Ellipses is better in case of students than the experts. The ellipses are considered to be one of the toughest to draw by the experts. The experts still try drawing these shapes at a faster speed which makes

Table 6.8: Features from subset selection

	Lines	Circles	Rectangles	Ellipses	Cuboids
Average deviation			✓		
Maximum deviation			✓		
Hausdorff similarity	✓				
Maximum speed					✓
Minimum speed	✓	✓	✓		✓
Average speed 1		✓	✓	✓	✓
Average speed 2	✓				✓
Maximum angle		✓	✓		✓
Average angle	✗		✗		
Average absolute angle			✓	✓	
Average square angle			✓		✓
Ratio of middle 50% to last 25% of stroke		✗			✓
Ratio of first 75% to last 25% of stroke	✓			✓	✗
Number of strokes			✓		
Stroke order			✓		
Stroke Direction				✓	
Ratio of path length to reference length				✓	✓
Ratio of average absolute angle to average speed		✓		✓	✓
Ratio of average square angle to average speed			✓	✓	✓
Ratio of average deviation to average speed		✓	✓	✓	✓
Ratio of maximum deviation to maximum speed			✓	✓	✓
Ratio of average deviation to size of shape	✓		✓		

their sketches significantly less accurate than the students. The gain ratio equal to 0.413 (highest among all the features) found in the subset selection method is evident that the speed is the most important feature in distinguishing between the two groups of users and experts are much faster than the students.

The accuracy is dependent on the size of the shape. Bigger the shapes, accuracy decreases because of the need in changing the hand gesture to draw bigger strokes. Pen can be moved over screen by moving either of wrist, elbow or shoulder. Lines, Circles and Rectangles have a reasonable distinction between the groups using this feature. The users cannot be distinguished using this feature for ellipses and cubes. While sketching ellipses and cuboids, the experts who understand perspective would try to achieve a perspective accuracy than just connecting dots like novices without knowledge about perspective would do.

The features that represent smoothness of strokes are better in experts than students. The average angle does not contribute in making the sketches of better quality. This is because average angle can be in both positive and negative quadrants, hence cancelling off when the stroke is wavy. Accuracy and smoothness together contribute to the visual clarity of the sketches. To see how they perform together, we had a feature which is product of the average deviation with average absolute angle and average square angle. This feature also performs better in case of experts.

Also, all the features used to measure the speed of the strokes have a very low p-value. The speed of the strokes are better in case of experts than the students. As explained in the previous chapter accuracy is affected by the speed with which one draws. It is expected that the faster one draws less is the accuracy. Though accuracy of experts and students is not significantly different, the ratio of average deviation and average speed is significantly different. This proves that when students draw at a speed equal to the experts their accuracy will be worse than the experts. We also checked the ratio of other visual features(accuracy and smoothness) to speed. All of these features performed well in dividing the sketches between the two sets of users we have in the study.

Both features to measure speed fluidity perform fairly well in case of Lines, Rectangles, Ellipses and Cubes. The t-values of these shapes for the features show that the speed decreases as pen approaches the end points in cases of straight lines and shapes that are made of straight lines, but increases for ellipses.

The features under planning perform moderately well. The number of strokes performs extremely well for rectangles, ellipses and cuboids. The experts draw with one stroke for each side of the rectangle and cuboid whereas students draw fewer strokes. For ellipses, experts try to draw the whole shape with one stroke but the students use multiple strokes to finish the shape. Ratio between the stroke length and the reference length and the stroke direction also perform well for our data set. Stroke order performs well in case of rectangles confirming that the experts draw the parallel lines together. For cuboids, the order of strokes does not help in distinguishing the users.

6.4 Classification

To test if the sets of feature actually can distinguish the experts and students, we perform Random forest on the data using 10-fold cross validation. We use Weka which is an open source package having a number of machine learning algorithms. The results of the classification are given in the table 6.9.

Table 6.9: Results of random forest classification

	Recall	Precision	F-measure
Lines	0.853	0.862	0.817
Circles	0.914	0.898	0.897
Rectangles	0.925	0.927	0.92
Ellipses	0.983	0.982	0.982
Cuboids	0.989	0.989	0.989

7. AUTOMATIC GRADING

We design a grading metric and grade the sketches using the set of features we finalised as explained in the previous chapter. We calculate the value for each of the features of the sketch and then get a final score of the sketch using weighted average of the scores.

Each feature is given a score between 0 and 1. For the features that have expert value more than the student value (for example, experts have speed features more than the novices) we give a score of 1 for any sketch that has value more than or equal to 30th percentile of the expert values. In another case of students having higher values than the experts (like deviation), a score of 0 is given to sketches that have more than 90th percentile of the student values and a score of 0 to feature value equal to 0. The values are linearly scaled with respect to these extreme values.

The final score is obtained by calculating a weighted average mean of the scores for individual features. The weights used for this calculations are obtained from the gain ratio analysis of the features. Gain ratio is calculated using multi-fold(k=10) cross validation where the set is divided into k sets randomly and one set is used for testing every time. This is done k times. The final score is in the range of 0 and 1. Figures 7.1, 7.2, 7.3, 7.4 and 7.5 show some of the high and low scoring sketches for each of the shapes. The equation 7.1 shows the calculation of scores of ellipse.

$$
\begin{aligned}
score = Ratio\ of\ path - length\ and\ reference\ length * 0.155+ \\
maximum\ speed * 0.158+ \\
minimum\ speed * 0.167+ \\
average\ speed * 0.366+ \\
average\ angle * 0.0+ \\
average\ absolute\ angle * 0.377+ \\
average\ square\ angle * 0.117+ \\
speed\ fluidity\ end2 * 0.007+ \\
speed\ Fluidity\ End * 0.077+ \\
no\ Of\ Strokes * 0.0+ \\
average\ absolute\ angle/average\ speed * 0.409+ \\
average\ square\ angle/average\ speed * 0.312+ \\
average\ deviation/average\ speed * 0.141+ \\
maximum\ deviation/maximum\ speed * 0.118+ \\
average\ speed2 * 0.413
\end{aligned}
\tag{7.1}
$$

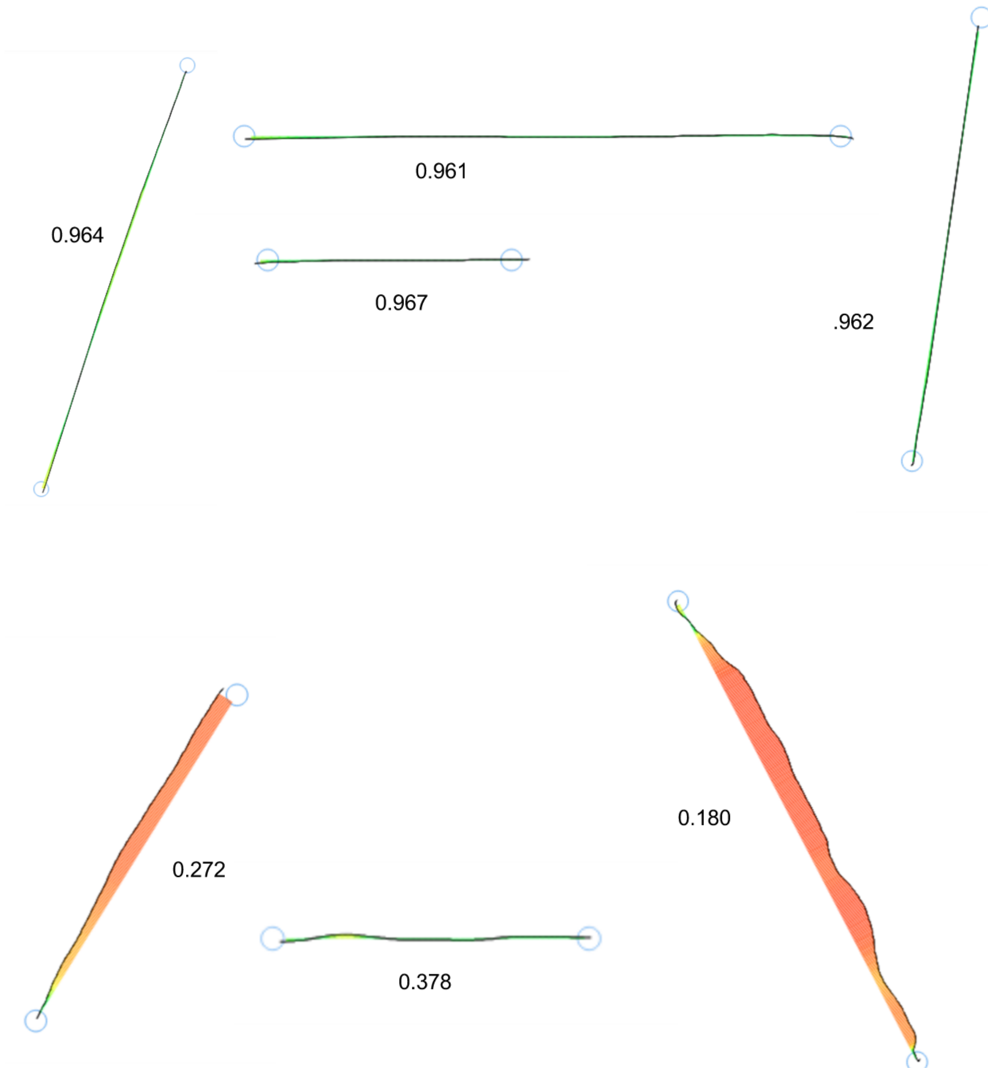

Figure 7.1: High score and low score lines

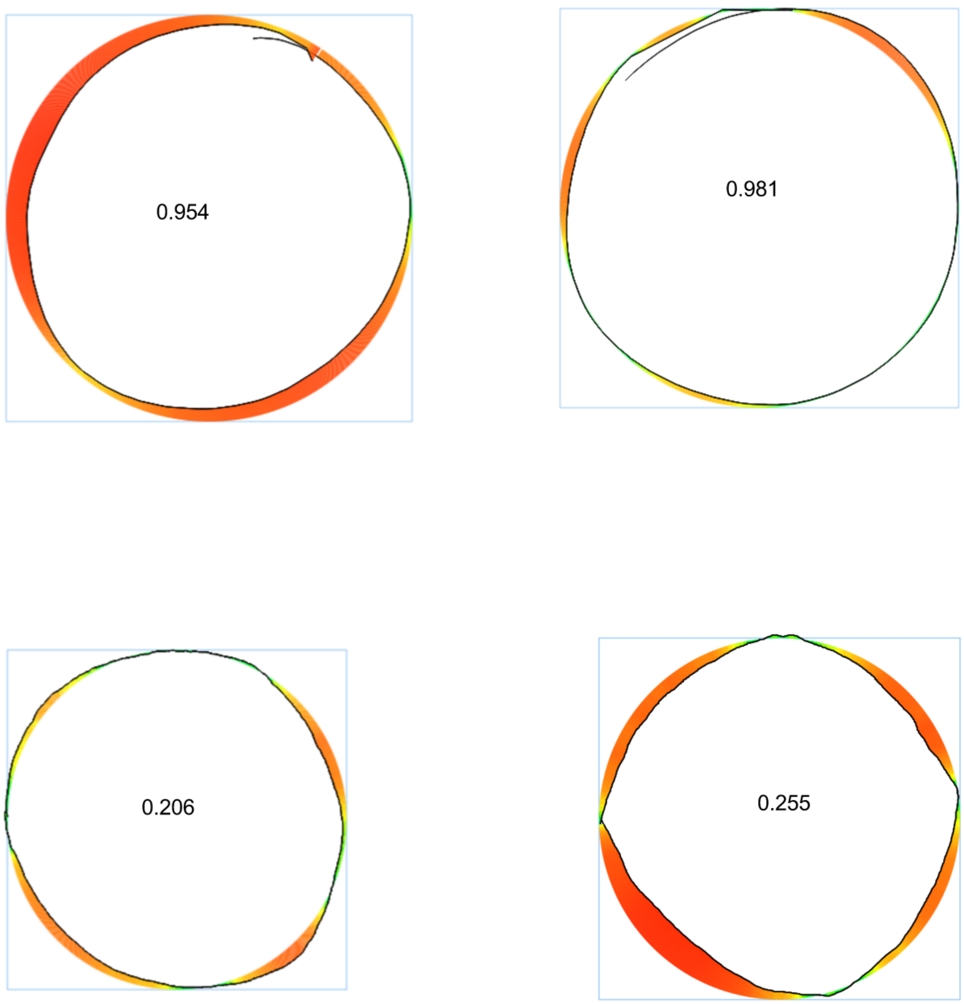

Figure 7.2: High score and low score circles

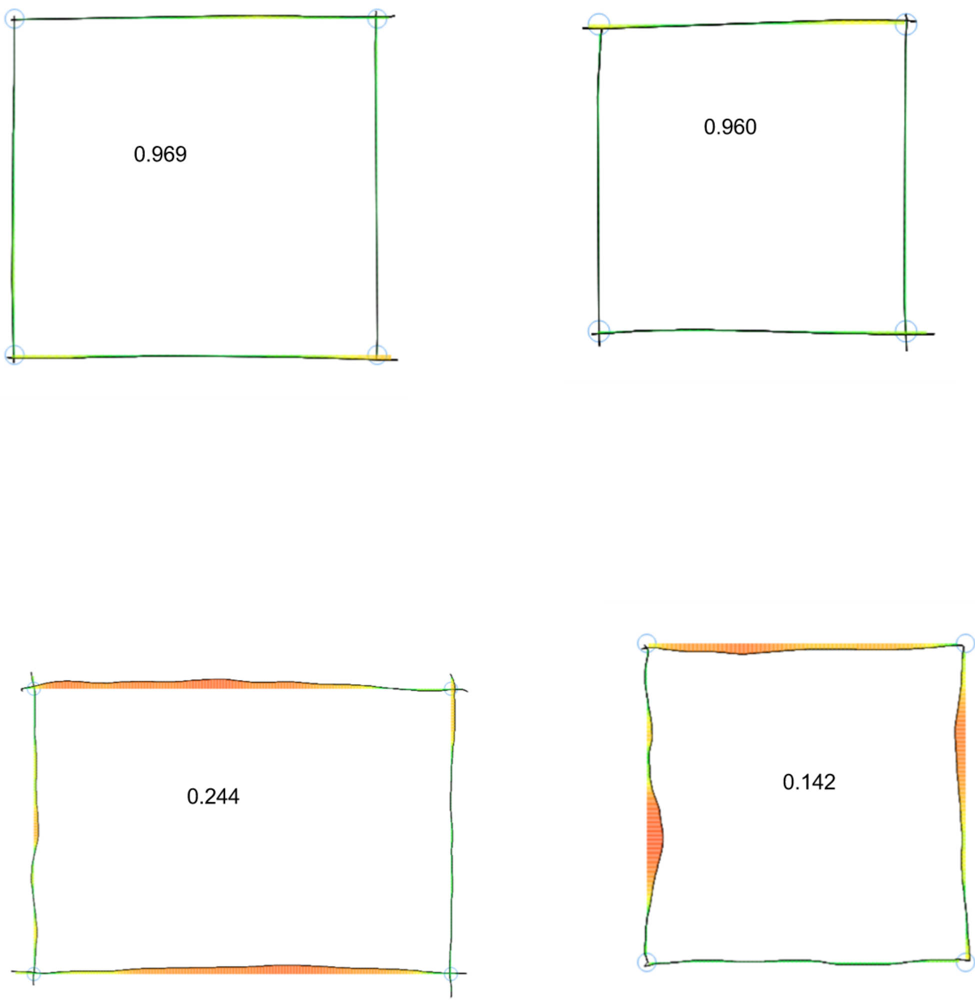

Figure 7.3: High score and low score rectangles

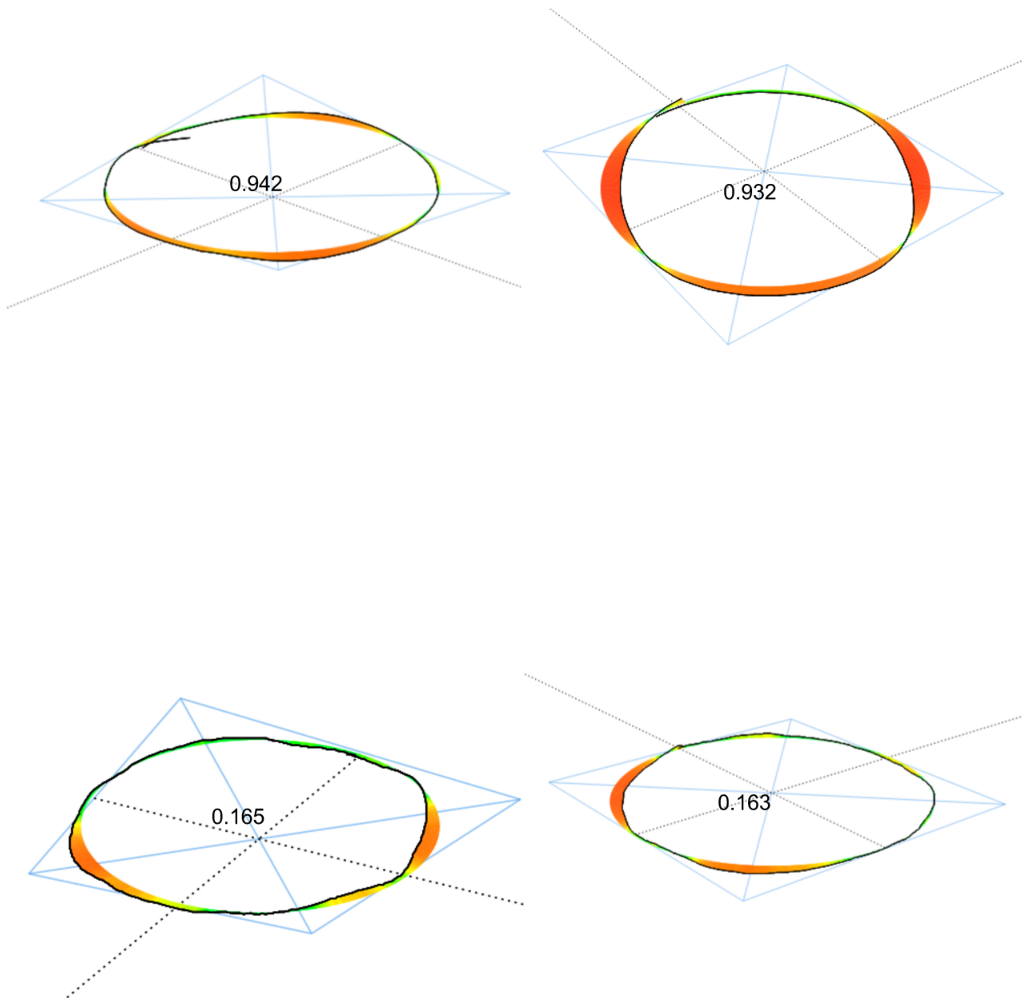

Figure 7.4: High score and low score ellipses

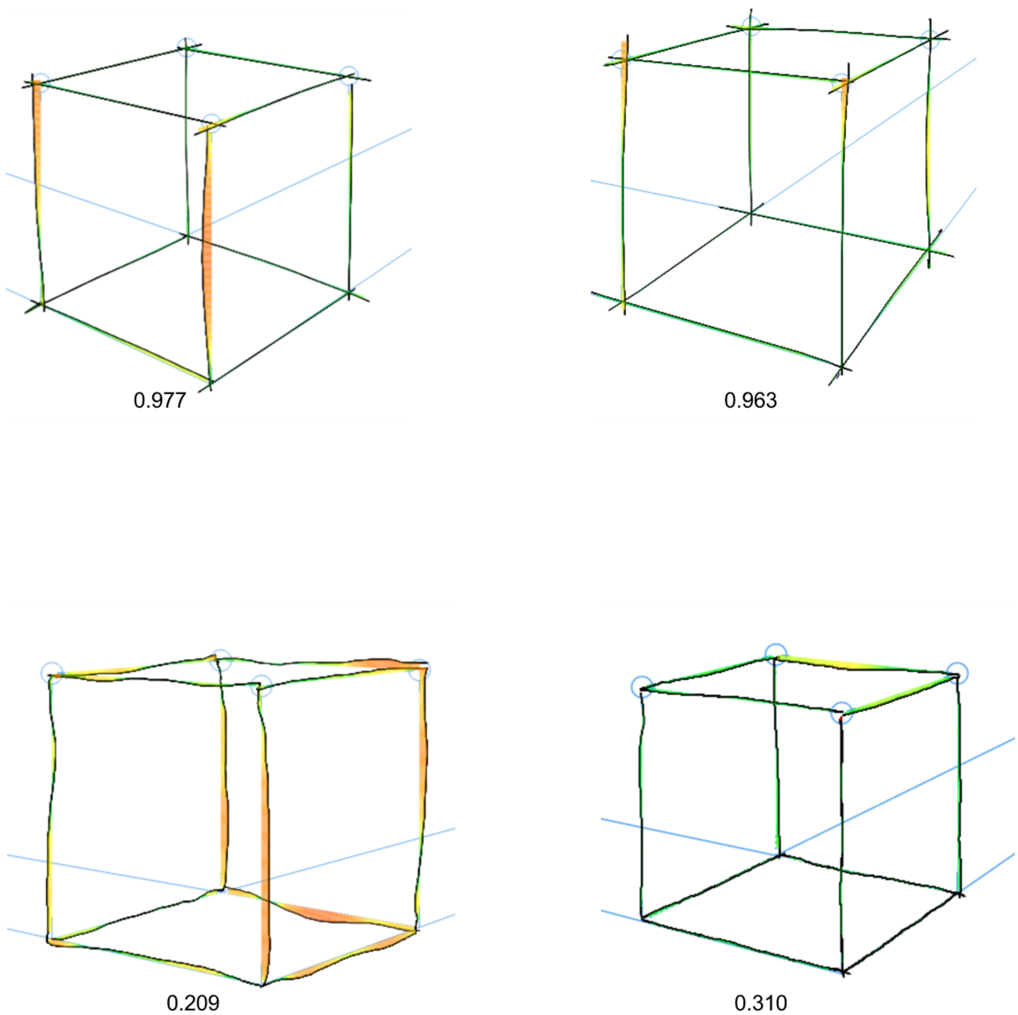

Figure 7.5: High score and low score cuboids

8. FUTURE WORK

We plan to implement a similar technique to grade student sketches for other basic and complex shapes as well. We also intend to create a more reliable grading system consistent with the actual perception of "good" quality sketches with respect to instructors and experts in this field. Our current approach is totally dependent on the computer to give a score. To evaluate the accuracy of this system with respect to the experts grading, we will get the sketches graded by multiple experts and the do a analysis to get a better grading system.

Pressure is also an important criteria in sketching to produce quality sketches. In future, a pressure sensitive device can be used to get pressure, line-weight relate features as well. Signature verification is a field that is similar to sketch recognition and user identification in PerSketchTivity. This field has had extensive research in the past, recognizing not only the distinct identity [24] of the sketcher, but also the age and gender of the sketcher [57, 58, 58]. Pressure and other features, such as entropy [7] should be tested to see what kind of benefits it gives to us in our study. Also, such algorithms can be used in implementing a sketch-based biometric system.

9. CONCLUSION

In this thesis, we designed a grading metric which can be used to automatically evaluate the design sketches. Design sketching is an important skill for engineering and design students. The feedback given in a traditional classroom set up is limited to instructor's availability and decreases the student's self-efficacy. We designed an intelligent-tutoring system to help students to learn sketching. From data collected from 20 students and 4 experts we performed statistical analysis to find the features important to distinguish between the two user groups for five shapes - lines, circles, ellipses, squares and cubes. The results for most of the features were positive with some features not performing well which have was not surprising on further thought and have been reasoned out. We used statistical analysis, subset selection for selecting features which can distinguish experts and students. We also performed classification on the data sample to understand how well we can classify using the features obtained. Using these features the objective values obtained from the sketch we designed a grading system which grades the students sketches between 0 and 1.

REFERENCES

[1] Underbeak. `http://goo.gl/EfVXDN`. Accessed: 2016-02-17.

[2] Johnny Accot and Shumin Zhai. Beyond fitts' law: Models for trajectory-based hci tasks. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, CHI '97, pages 295–302, New York, NY, USA, 1997. ACM.

[3] Olufunmilola Atilola, Martin Field, Erin McTigue, Tracy Hammond, and Julie Linsey. Mechanix: a sketch recognition truss tutoring system. In *ASME 2011 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages 645–654, Washington, DC, USA, 2011. American Society of Mechanical Engineers.

[4] Olufunmilola Atilola, Stephanie Valentine, Hong-Hoe Kim, David Turner, Erin McTigue, Tracy Hammond, and Julie Linsey. Mechanix: A natural sketch interface tool for teaching truss analysis and free-body diagrams. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 28:169–192, 5 2014.

[5] Robert L. Bangert-Drowns, Chen-Lin C. Kulik, James A. Kulik, and Mary-Teresa Morgan. The instructional effect of feedback in test-like events. *Review of Educational Research*, 61(2):213–238, 1991.

[6] Luca Benedetti, Holger Winnemöller, Massimiliano Corsini, and Roberto Scopigno. Painting with bob: Assisted creativity for novices. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST '14, pages 419–428, New York, NY, USA, 2014. ACM.

[7] Akshay Bhat and Tracy Hammond. Using entropy to distinguish shape versus text in hand-drawn diagrams. In *Proceedings of the 21st International Joint Conference on Artifical Intelligence*, IJCAI'09, pages 1395–1400, San Francisco, CA, USA, 2009. Morgan Kaufmann Publishers Inc.

[8] Mary E Broadbent, Martin Brown, Kevin Penner, I Ipsen, and Rizwana Rehman. Subset selection algorithms: Randomized vs. deterministic. *SIAM Undergraduate Research Online*, 3:50–71, 2010.

[9] Bill Buxton. *Sketching User Experiences: Getting the Design Right and the Right Design*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2007.

[10] Hsiang Chen, R.T. Wigand, and M.S. Nilan. Optimal experience of web activities. *Computers in Human Behavior*, 15(5):585 – 608, 1999.

[11] Heeyoul Choi and Tracy Hammond. Sketch recognition based on manifold learning. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 3*, AAAI'08, pages 1786–1787, Chicago, Illinois, USA, 2008. AAAI Press.

[12] Heeyoul Choi, Brandon Paulson, and Tracy Hammond. Gesture recognition based on manifold learning. In *Proceedings of the 2008 Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition*, SSPR & SPR '08, pages 247–256, Berlin, Heidelberg, 2008. Springer-Verlag.

[13] Paul Corey and Tracy Hammond. Gladder: Combining gesture and geometric sketch recognition. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 3*, AAAI'08, pages 1788–1789, Chicago, Illinois, 2008. AAAI Press.

[14] Danielle Cummings, Stephane Fymat, and Tracy Hammond. Sketch-based interface for interaction with unmanned air vehicles. In *CHI '12 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '12, pages 1511–1516, New York, NY, USA, 2012. ACM.

[15] Danielle Cummings, Francisco Vides, and Tracy Hammond. I don't believe my eyes!: Geometric sketch recognition for a computer art tutorial. In *Proceedings of the International Symposium on Sketch-Based Interfaces and Modeling*, SBIM '12, pages 97–106, Aire-la-Ville, Switzerland, Switzerland, 2012. Eurographics Association.

[16] D. Cummmings, S. Fymat, and T. Hammond. Reddog: A smart sketch interface for autonomous aerial systems. In *Proceedings of the International Symposium on Sketch-Based Interfaces and Modeling*, SBIM '12, pages 21–28, Aire-la-Ville, Switzerland, Switzerland, 2012. Eurographics Association.

[17] Katie Dahmen and Tracy Hammond. Distinguishing between sketched scribble look alikes. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 3*, AAAI'08, pages 1790–1791, Chicago, Illinois, USA, 2008. AAAI Press.

[18] Karen Ernst daSilva. Drawing on experience: Connecting art and language. *Primary Voices K-6*, 10(2):2–9, 2001.

[19] D. Davies and A. White. *Harrap's Illustrated Dictionary of Art & Artists.* Harrap's Reference, 1990.

[20] Ruwanee de Silva, David Tyler Bischel, WeeSan Lee, Eric J. Peterson, Robert C. Calfee, and Thomas F. Stahovich. Kirchhoff's pen: A pen-based circuit analysis tutor. In *Proceedings of the 4th Eurographics Workshop on*

Sketch-based Interfaces and Modeling, SBIM '07, pages 75–82, New York, NY, USA, 2007. ACM.

[21] Daniel Dixon, Manoj Prasad, and Tracy Hammond. icandraw: Using sketch recognition and corrective feedback to assist a user in drawing human faces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI 10, pages 897–906, New York, New York, USA, 2010. ACM.

[22] Ronald G Ehrenberg, Dominic J Brewer, Adam Gamoran, and J Douglas Willms. Class size and student achievement. *Psychological Science in the Public Interest*, 2(1):1–30, 2001.

[23] Koos Eissen and Roselien Steur. *Sketching: Drawing Techniques for Product Designers*. BIS Publishers, Amsterdam, The Netherlands, 12 edition, 2009.

[24] Brian David Eoff and Tracy Hammond. Who dotted that 'i'?: Context free user differentiation through pressure and tilt pen data. In *Proceedings of Graphics Interface 2009*, GI '09, pages 149–156, Toronto, Ont., Canada, Canada, 2009. Canadian Information Processing Society.

[25] Martin Field, Stephanie Valentine, Julie Linsey, and Tracy Hammond. Sketch recognition algorithms for comparing complex and unpredictable shapes. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Three*, IJCAI'11, pages 2436–2441. AAAI Press, 2011.

[26] Paul M Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47(6):381, 1954.

[27] Tamar Flash and Neville Hogan. The coordination of arm movements: an experimentally confirmed mathematical model. *The Journal of Neuroscience*, 5(7):1688–1703, 1985.

[28] Leslie Gennari, Levent Burak Kara, Thomas F Stahovich, and Kenji Shimada. Combining geometry and domain knowledge to interpret hand-drawn diagrams. *Computers & Graphics*, 29(4):547–562, 2005.

[29] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The weka data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009.

[30] T Hammond and R Davis. Creating the perception-based ladder sketch recognition language. In *Proceedings of the 8th ACM Conference on Designing Interactive Systems*, DIS '10, pages 141–150, New York, NY, USA, 2010. ACM.

[31] Tracy Hammond. Automatically generating sketch interfaces from shape descriptions. In *Proceedings of the 4th Annual MIT Student Oxygen Workshop*, page 4. MIT, 2004.

[32] Tracy Hammond. *Ladder: A perceptually-based language to simplify sketch recognition user interface development*. PhD thesis, Massachusetts Institute of Technology, 2007.

[33] Tracy Hammond. *Sketch Recognition: Algorithms and Applications*. Cambridge University Press, 2017. draft from March 1, 2016, publication forthcoming.

[34] Tracy Hammond and Randall Davis. Ladder: A language to describe drawing, display, and editing in sketch recognition. In *Proceedings of the International Joint Conference on Aritificial Intelligence (IJCAI)*, pages 461–467, 2003.

[35] Tracy Hammond and Randall Davis. Automatically transforming symbolic shape descriptions for use in sketch recognition. In *Proceedings of the 19th National Conference on Artifical Intelligence*, AAAI'04, pages 450–456. AAAI Press, 2004.

[36] Tracy Hammond and Randall Davis. Shady: A shape description debugger for use in sketch recognition. In *AAAI Fall Symposium on Making Pen-Based Interaction Intelligent and Natural*, Arlington, VA, October 2004. AAAI. 7 pages.

[37] Tracy Hammond and Randall Davis. Testing shape descriptions by automatically translating them for use in sketch recognition. In *MIT Lab Abstract*, page 2. MIT, 2004.

[38] Tracy Hammond and Randall Davis. Ladder, a sketching language for user interface developers. In *Computers & Graphics*, volume 29:4, pages 518–532. Elsevier, 2005.

[39] Tracy Hammond and Randall Davis. Interactive learning of structural shape descriptions from automatically generated near-miss examples. In *Proceedings of the 11th International Conference on Intelligent User Interfaces*, IUI '06, pages 210–217, New York, NY, USA, 2006. ACM.

[40] Tracy Hammond and Randall Davis. Tahuti: A geometrical sketch recognition system for uml class diagrams. In *ACM SIGGRAPH 2006 Courses*, SIGGRAPH '06, New York, NY, USA, 2006. ACM.

[41] Tracy Hammond, Brian Eoff, Brandon Paulson, Aaron Wolin, Katie Dahmen, Joshua Johnston, and Pankaj Rajan. Free-sketch recognition: Putting the chi in sketching. In *CHI '08 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '08, pages 3027–3032, New York, NY, USA, 2008. ACM.

[42] Tracy Hammond, Krzysztof Gajos, Randall Davis, and Howard E Shrobe. An agent-based system for capturing and indexing software design meetings. In *In Proceedings of the International Workshop on Agents in Design (WAID02*, volume 2, pages 203–218, 2002.

[43] Tracy Hammond, Drew Logsdon, Joshua Peschel, Joshua Johnston, Paul Taele, Aaron Wolin, and Brandon Paulson. A sketch recognition interface that recognizes hundreds of shapes in course-of-action diagrams. In *CHI '10 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '10, pages 4213–4218, New York, NY, USA, 2010. ACM.

[44] Tracy Hammond and Brandon Paulson. Recognizing sketched multistroke primitives. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 1(1):1–34, 2011.

[45] Tracy Hammond, Manoj Prasad, and Daniel Dixon. Art 101: Learning to draw through sketch recognition. In *Proceedings of the 10th International Conference on Smart Graphics*, SG'10, pages 277–280, Berlin, Heidelberg, 2010. Springer-Verlag.

[46] Tracy Anne Hammond, Drew Logsdon, Brandon Paulson, Joshua Johnston, Joshua Peschel, Aaron Wolin, and Paul Taele. A sketch recognition system for recognizing free-hand course of action diagrams. In *Twenty-Second IAAI Conference*, pages 1781–1786, July 11-15 2010.

[47] Jay D. Helsel. *Reading Engineering Drawings Through Conceptual Sketching*. Glencoe/Mcgraw-Hill, Columbus, Ohio, USA, 1 edition, 1979.

[48] Richard Hickman, Sarah Lord, et al. An examination of adolescents' self-efficacy, engagement and achievement in representational drawing. *Australian Art Education*, 32(2):73, 2009.

[49] Kelly Hodgkins. Daily ipad app: Draw this app challenges you to refine your drawing technique. `http://goo.gl/RX57sn`, 2013. Accessed: 2016-01-20.

[50] CI Howarth, WDA Beggs, and JM Bowden. The relationship between speed and accuracy of movement aimed at a target. *Acta Psychologica*, 35(3):207–218, 1971.

[51] Emmanuel Iarussi, Adrien Bousseau, and Theophanis Tsandilas. The drawing assistant: Automated drawing guidance and feedback from photographs. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, UIST '13, pages 183–192, New York, New York, USA, 2013. ACM.

[52] J. Johnston and T. Hammond. Computing confidence values for geometric constraints for use in sketch recognition. In *Proceedings of the Seventh Sketch-Based Interfaces and Modeling Symposium*, SBIM '10, pages 71–78, Aire-la-Ville, Switzerland, Switzerland, 2010. Eurographics Association.

[53] E.E. Jones. The correlation of visual memory and perception of perspective with drawing ability. *School and Society*, 15:174–176, 1922.

[54] Jr. Joseph J. LaViola and Robert C. Zeleznik. Mathpad2: A system for the creation and exploration of mathematical sketches. In *ACM SIGGRAPH 2004 Papers*, SIGGRAPH 2004, pages 432–440, New York, New York, USA, 2004. ACM.

[55] Levent Burak Kara and Thomas F Stahovich. An image-based, trainable symbol recognizer for hand-drawn sketches. *Computers & Graphics*, 29(4):501–517, 2005.

[56] Kourtney Kebodeaux, Martin Field, and Tracy Hammond. Defining precise measurements with sketched annotations. In *Proceedings of the Eighth Eurographics Symposium on Sketch-Based Interfaces and Modeling*, SBIM '11, pages 79–86, New York, NY, USA, 2011. ACM.

[57] Hong-Hoe Kim, Paul Taele, Stephanie Valentine, Jeff Liew, and Tracy Hammond. Developing intelligent sketch-based applications to support children's self-regulation and school readiness. In *2014 Intelligent User Interfaces Workshop on Sketch Recognition*, pages 1–8. ACM, 2014.

[58] Hong-Hoe Kim, Stephanie Valentine, Paul Taele, and Tracy Hammond. Easysketch: A sketch-based educational interface to support childrens self-regulation and school readiness. In *The Impact of Pen and Touch Technology on Education*, Human–computer Interaction Series, pages 35–46. Springer, 2015.

[59] Wenzhe Li and Tracy Hammond. Using scribble gestures to enhance editing behaviors of sketch recognition systems. In *CHI '12 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '12, pages 2213–2218, New York, NY, USA, 2012. ACM.

[60] George Lucchese, Martin Field, Jimmy Ho, Ricardo Gutierrez-Osuna, and Tracy Hammond. Gesturecommander: Continuous touch-based gesture prediction. In *CHI '12 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '12, pages 1925–1930, New York, NY, USA, 2012. ACM.

[61] Nic Lupfer, Martin Field, Andruid Kerne, and Tracy Hammond. sketchy: Morphing user sketches for artistic assistance. In *2011 Intelligent User Interfaces Workshop on Sketch Recognition*, page 4, Palo Alto, CA, February 13-16 2011. ACM.

[62] Michael Helms Julie S. Linsey Matthew G. Green, Benjamin W. Caldwell and Tracy Anne Hammond. Using natural sketch recognition software to provide instant feedback on statics homework (truss free body diagrams): Assessment of a classroom pilot. In *2015 ASEE Annual Conference and Exposition*, number 10.18260/p.25007, Seattle, Washington, June 2015. ASEE Conferences. https://peer.asee.org/25007.

[63] Inc MEDL Mobile. Learn to draw sketchbook by walter foster. `https://goo.gl/LdnV4n`. Accessed: 2016-01-20.

[64] David E Meyer, JE Keith-Smith, Sylvan Kornblum, Richard A Abrams, and Charles E Wright. Speed-accuracy tradeoffs in aimed movements: Toward a theory of rapid voluntary action. *Attention and Performance 13: Motor Representation and Control*, pages pp. 173–226, 1990.

[65] Erik G Miller, Nicholas E Matsakis, and Paul A Viola. Learning from one example through shared densities on transforms. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 1, pages 464–471. IEEE, 2000.

[66] Trevor Nelligan, Seth Polsley, Jaideep Ray, Michael Helms, Julie Linsey, and Tracy Hammond. Mechanix: A sketch-based educational interface. In *Proceedings of the 20th International Conference on Intelligent User Interfaces Companion*, IUI Companion '15, pages 53–56, New York, NY, USA, 2015. ACM.

[67] Luke Olsen, Faramarz F Samavati, Mario Costa Sousa, and Joaquim A Jorge. Sketch-based modeling: A survey. *Computers & Graphics*, 33(1):85–103, 2009.

[68] OMGPOP. Draw something. `https://goo.gl/GiS3hl`. Accessed: 2016-01-20.

[69] Jane Parker. A consideration of the relationship between creativity and approaches to learning in art and design. *International Journal of Art & Design Education*, 24(2):186–198, 2005.

[70] Brandon Paulson, Brian Eoff, Aaron Wolin, Joshua Johnston, and Tracy Hammond. Sketch-based educational games: Drawing kids away from traditional interfaces. In *Proceedings of the 7th International Conference on Interaction Design and Children*, IDC '08, pages 133–136, New York, NY, USA, 2008. ACM.

[71] Brandon Paulson and Tracy Hammond. Paleosketch: Accurate primitive sketch recognition and beautification. In *Proceedings of the 13th International Conference on Intelligent User Interfaces*, IUI '08, pages 1–10, New York, NY, USA, 2008. ACM.

[72] Joshua M Peschel and Tracy Anne Hammond. Strat: A sketched-truss recognition and analysis tool. In *2008 International Workshop on Visual Languages and Computing (VLC) at the 14th International Conference on distributed Multimedia Systems (DMS)*, page 282287, Boston, MA, 9 2008. Knowledge Systems Institute.

[73] Manoj Prasad and Tracy Hammond. Observational study on teaching artifacts created using tablet pc. In *CHI '12 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '12, pages 301–316, New York, NY, USA, 2012. ACM.

[74] Pankaj Rajan and T. Hammond. From paper to machine: Extracting strokes from images for use in sketch recognition. In *Proceedings of the Fifth Eurographics Conference on Sketch-Based Interfaces and Modeling*, SBM'08, pages 41–48, Aire-la-Ville, Switzerland, Switzerland, 2008. Eurographics Association.

[75] Dwayne Raymond, Jeffrey Liew, and Tracy A Hammond. A vision for education: Transforming how formal systems are taught within mass lectures by using pen technology to create a personalized learning environment. In Tracy Hammond, Stephanie Valentine, Aaron Adler, and Mark Payton, editors, *The Impact of Pen and Touch Technology on Education*, pages 355–363. Springer International Publishing Switzerland, 2015.

[76] Dan Roam. *The Back of the Napkin: Solving Problems and Selling Ideas with Pictures*. Portfolio Publishing, London, England, United Kingdom, 1 edition, 2013.

[77] Dean Rubine. Specifying gestures by example. In *Proceedings of the 18th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '91, pages 329–337, New York, NY, USA, 1991. ACM.

[78] Irene Schiferl. Both sides now: Visualizing and drawing with the right and left hemispheres of the brain. *Studies in Art Education*, 50(1):67–82, 2008.

[79] Tevfik Metin Sezgin and Randall Davis. Scale-space based feature point detection for digital ink. In *ACM SIGGRAPH 2007 Courses*, SIGGRAPH '07, New York, NY, USA, 2007. ACM.

[80] Tevfik Metin Sezgin, Thomas Stahovich, and Randall Davis. Sketch based interfaces: Early processing for sketch understanding. In *ACM SIGGRAPH 2006 Courses*, SIGGRAPH '06, New York, NY, USA, 2006. ACM.

[81] HerbertA. Simon. Why should machines learn? In RyszardS. Michalski, JaimeG. Carbonell, and TomM. Mitchell, editors, *Machine Learning*, Symbolic Computation, pages 25–37. Springer Berlin Heidelberg, 1983.

[82] K. Sjölén and A. MacDonald. *Learning Curves: An Inspiring Guide to Improve Your Design Sketch Skills*. KEEOS Design Books, 2011.

[83] Sheryl Sorby. Educational research in developing 3d spatial skills for engineering students. *International Journal of Science Education*, 31(3):459–480, Feb 2009.

[84] Thomas F Stahovich. Segmentation of pen strokes using pen speed. In *AAAI Fall Symposium Series*, pages 21–24, 2004.

[85] Underbeak Studios. Circled. `https://goo.gl/cyMvdi`. Accessed: 2016-01-28.

[86] Paul Taele, Laura Barreto, and Tracy Hammond. Hashigo: A next-generation sketch interactive system for japanese kanji. In *Proceedings of the Twenty-First Innovative Applications of Artificial Intelligence Conference*, IAAI '15, pages 153–158, Palo Alto, California, USA, 2009. AAAI.

[87] Paul Taele, Laura Barreto, and Tracy Hammond. Maestoso: An intelligent educational sketching tool for learning music theory. In *Proceedings of the Twenty-Seventh Innovative Applications of Artificial Intelligence Conference*, IAAI '15, pages 3999–4005, Palo Alto, California, USA, 2015. AAAI.

[88] Paul Taele and Tracy Hammond. Using a geometric-based sketch recognition approach to sketch chinese radicals. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 3*, AAAI'08, pages 1832–1833. AAAI Press, 2008.

[89] Paul Taele and Tracy Hammond. Lamps: A sketch recognition-based teaching tool for mandarin phonetic symbols i. *J. Vis. Lang. Comput.*, 21(2):109–120, April 2010.

[90] Paul Taele and Tracy Hammond. Enhancing instruction of written east asian languages with sketch recognition-based intelligent language workbook interfaces. In *The Impact of Pen and Touch Technology on Education*, pages 119–126. Springer Publishing Company, Incorporated, 1st edition, 2015.

[91] Barbara Tversky. Visualizing thought. *Topics in Cognitive Science*, 3(3):499–535, 2011.

[92] David G Ullman, Stephen Wood, and David Craig. The importance of drawing in the mechanical design process. *Computers & graphics*, 14(2):263–274, 1990.

[93] Stephanie Valentine, Martin Field, Anne Smith, and Tracy Hammond. A shape comparison technique for use in sketch-based tutoring systems. In *2011 Intelligent User Interfaces Workshop on Sketch Recognition*, page 4. IUI, 2011.

[94] Stephanie Valentine, Raniero Lara-Garduno, Julie Linsey, and Tracy Hammond. Mechanix: A sketch-based tutoring system that automatically corrects hand-sketched statics homework. In *The Impact of Pen and Touch Technology on Education*, pages 91–105. Springer Publishing Company, Incorporated, 1st edition, 2015.

[95] Stephanie Valentine, Francisco Vides, George Lucchese, David Turner, Hong hoe Kim, Wenzhe Li, Julie Linsey, and Tracy Hammond. Mechanix: A sketch-based tutoring system for statics courses. In *Proceedings of the Twenty-Fourth Innovative Applications of Artificial Intelligence Conference*, IAAI '12, pages 2253–2260, Palo Alto, California, USA, 2012. AAAI.

[96] Stephanie Valentine, Francisco Vides, George Lucchese, David Turner, Hong-Hoe Kim, Wenzhe Li, Julie Linsey, and Tracy Hammond. Mechanix: a sketch-based tutoring and grading system for free-body diagrams. *AI Magazine*, 34(1):55–66, 2013.

[97] Francisco Vides, Paul Taele, Hong-Hoe Kim, Jimmy Ho, and Tracy Hammond. Intelligent feedback for kids using sketch recognition. In *ACM SIGCHI 2012 Conference on Human Factors in Computing Systems Workshop on Educational Interfaces, Software, and Technology*, Austin, TX, USA, 2012. ACM.

[98] A.T. Welford, A.H. Norris, and N.W. Shock. Speed and accuracy of movement and their changes with age. *Acta Psychologica*, 30:3 – 15, 1969.

[99] Jacob O. Wobbrock, Andrew D. Wilson, and Yang Li. Gestures without libraries, toolkits or training: A $1 recognizer for user interface prototypes. In *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology*, UIST '07, pages 159–168, New York, NY, USA, 2007. ACM.

[100] A. Wolin, B. Paulson, and T. Hammond. Sort, merge, repeat: An algorithm for effectively finding corners in hand-sketched strokes. In *Proceedings of the 6th Eurographics Symposium on Sketch-Based Interfaces and Modeling*, SBIM '09, pages 93–99, New York, NY, USA, 2009. ACM.

[101] Aaron Wolin, Brian Eoff, and Tracy Hammond. Shortstraw: A simple and effective corner finder for polylines. In *Proceedings of the Fifth Eurographics Conference on Sketch-Based Interfaces and Modeling*, SBIM'08, pages 33–40, Aire-la-Ville, Switzerland, Switzerland, 2008. Eurographics Association.

[102] Aaron Wolin, Martin Field, and Tracy Hammond. Combining corners from multiple segmenters. In *Proceedings of the Eighth Eurographics Symposium on Sketch-Based Interfaces and Modeling*, SBIM '11, pages 117–124, New York, NY, USA, 2011. ACM.

[103] Aaron Wolin, Brandon Paulson, and Tracy Hammond. Eliminating false positives during corner finding by merging similar segments. In *Proceedings of*

the 23rd National Conference on Artificial Intelligence - Volume 3, AAAI'08, pages 1836–1837. AAAI Press, 2008.

[104] Jun Xie, Aaron Hertzmann, Wilmot Li, and Holger Winnemöller. Portraitsketch: Face sketching assistance for novices. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST '14, pages 407–417, New York, New York, USA, 2014. ACM.

[105] Shumin Zhai, Jing Kong, and Xiangshi Ren. Speed-accuracy tradeoff in fitts' law tasks: On the equivalency of actual and nominal pointing precision. *International Journal of Human-Computer Studies*, 61(6):823–856, December 2004.